Universidad Zaragoza

Ingeniería de Tecnologías y Servicios de Telecomunicación

Curso 2019-2020

Trabajo Fin de Grado

Deep Generative Models para sensores acústicos distribuidos

Antonio Almudévar Atienza

Tutor

Alfonso Ortega Giménez

2019-2020

RESUMEN

En este texto se presenta una solución que permite detectar eventos en sistemas de sensado acústico distribuido (DAS) [1] haciendo uso de sistemas de aprendizaje profundo. La idea del DAS es relativamente innovadora y tiene variedad de aplicaciones, algo que motiva el desarrollo de trabajos como este. La forma de abordar el problema aquí considera los eventos como sucesos atípicos, por lo que se parte de un enfoque basado en técnicas de detección de anomalías, algo que supone una manera innovadora de abordar este problema. En cuanto a la forma de llevar a cabo la implementación de un sistema de detección de anomalías, se ha optado por el uso de técnicas de aprendizaje profundo debido a la flexibilidad que proporcionan y que permite adaptar todo lo aquí explicado a entornos físicos diferentes manteniendo la misma idea conceptual.

Concretamente, este es un proyecto lanzado por la empresa Aragón Photonics y es una tecnología novedosa a nivel mundial [2]. Precisamente, al ser un avance reciente no existe apenas literatura sobre el tema y la solución que aquí se propone parte desde cero. Es por esto, que este trabajo es realmente una aproximación a lo que podría ser un sistema final basado en algunos de los principios que aquí se presentan. Independientemente de esto, la solución que se propone es completamente funcional y da los resultados esperados y podría ser fácilmente adaptada a entornos distintos al que se han realizado las pruebas.

Cabe destacar también, que la caracterización de algunos de los factores que describen las prestaciones del sistema de cara a probabilidades de acierto o fallo en una detección se ha visto perjudicada por la situación coyuntural y extraordinaria que ha tocado vivir durante el desarrollo de este trabajo. Es por esto que el funcionamiento del sistema podría ser mejor caracterizado y medido y además cabría la posibilidad de, empleando la misma estructura que aquí se propone, mejorar el rendimiento del mismo en el caso de contar con un banco de medidas claramente más amplio

Por último, en cuanto a la distribución de este texto, se dividirá en varios capítulos. En el primero se tratará de explicar el problema que se plantea al comienzo del trabajo y los objetivos. El segundo será una explicación de conceptos teóricos que son necesarios para la comprensión del trabajo. El tercero será la explicación de lo que se ha realizado y de cada una de las partes que conforman el sistema. En el cuarto se comentarán los resultados experimentales y el quinto, finalmente, tratará las conclusiones que se pueden extraer de este proyecto y líneas futuras de trabajo.

AGRADECIMIENTOS

Me gustaría empezar acordándome de mis padres, ya que es a ellos a quienes les debo prácticamente todo lo que tengo. También por su apoyo incondicional, especialmente en estos meses de trabajo y confinamiento. También, me gustaría agradecer al resto de mi familia el hecho de preocuparse por mí y por lo que hago.

Por otro lado, me gustaría dar las gracias a mis amigos de siempre, en los que sé que puedo confiar cuando los necesito. También, a todas las personas que he conocido estos últimos cuatro años y que me han acompañado en este bonito camino.

Por último, pero no por ello menos importante, me gustaría agradecer a los profesores con los que me he encontrado a lo largo de mi vida académica. En concreto, me gustaría agradecer a Alfonso toda la ayuda y atención prestada estos últimos meses y el hecho de haber confiado en este proyecto.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1. INTRODUCCIÓN

En este capítulo se va a tratar de plantear el problema, la situación antes de comenzar este proyecto y algunas aplicaciones que tienen los sistemas de sensado acústico distribuido.

1.1. Motivación del proyecto

En los últimos años se han presentado técnicas que emplean fenómenos físicos que se dan en la propagación de ondas electromagnéticas, en particular de la luz, para detectar eventos de tipo mecánico en el entorno físico de una fibra óptica. Estas técnicas se basan principalmente en el impacto que un evento de tipo mecánico tiene en el fenómeno de la dispersión Rayleigh, algo que se explicará un poco más en detalle en el siguiente capítulo. Los sistemas que desarrollan estas técnicas se conocen como DAS (distributed acoustic sensor) [1], ya que emplean la fibra óptica como sensor distribuido para captar eventos en sus proximidades haciendo uso de transmisores y receptores ópticos.

El porqué de la relevancia de un sistema capaz de detectar eventos en el entorno de la fibra óptica es simplemente la multitud de aplicaciones que esto puede tener, desde seguridad [3] hasta control de tráfico. Desde un punto de la seguridad, este sistema podría suponer un método para detectar un vehículo circulando por una zona prohibida o a una persona picando en un terreno bajo el cual hay una fibra óptica. De manera similar, no es difícil darse cuenta que teniendo una medida que contiene información sobre la evolución de un cuerpo en tiempo y distancia, se puede calcular la velocidad del mismo. También se puede detectar una acumulación de vehículos en una zona concreta, lo que puede ser un indicativo de que se ha producido un accidente. Y todo esto se puede conseguir con un sistema invisible a los ojos del usuario, lo cual es una ventaja fundamental de esta idea. También, esta tecnología se ha utilizado, por ejemplo, para detectar actividad sísmica [4, 5].

Otra ventaja fundamental es el aprovechamiento de lo que se conoce como fibra oscura, esto es, fibra óptica que ha sido desplegada y a la que no se la ha dado ningún uso. La razón de ser de esta fibra oscura es precisamente facilitar la implementación de técnicas que en un principio no habían sido planteadas sin suponer esto un gran coste económico ni la necesidad de solicitar gran cantidad de permisos burocráticos [6].

1.2. Introducción a la problemática

Este trabajo, en cambio, no tiene como objetivo estudiar y desarrollar sistemas basados en estos fenómenos sino más bien trabajar con las señales capturadas por receptores que funcionan gracias a esta tecnología y tratar de encontrar en ellas los cambios que un evento produce en el comportamiento habitual de la señal óptica propagada dentro de la fibra. Hasta el momento, existen técnicas cuyo funcionamiento ha sido contrastado para detectar estos eventos, pero en este trabajo se propone una idea innovadora para solucionar este problema y que está basada en la utilización de sistemas de aprendizaje profundo.

La señal bidimensional (las dimensiones son tiempo y frecuencia) procedente de los receptores se muestra ruidosa y sin ningún tratamiento resulta difícil de interpretar. Hasta el momento se había recurrido a soluciones basadas en filtrado adaptativo, pero en este trabajo se propone dar un paso más y emplear métodos de aprendizaje automático no supervisado. Esto, en otras palabras, significa diseñar una red neuronal que pueda ser capaz por ella misma, sin la ayuda de un humano, de detectar un evento que consideraremos anómalo en las inmediaciones de la fibra a partir de la señal proporcionada por los receptores acústicos. El hecho de abordar el problema de un modo no supervisado es ideal porque el procedimiento conceptual es exactamente el mismo para cualquier entorno y lo aplicado con el banco de pruebas que conforma la base de datos que ha servido para mostrar los resultados obtenidos podría ser empleado sin apenas cambios en un entorno completamente distinto, ya que podemos ver este trabajo como una forma de detectar anomalías en una señal de dos dimensiones.

1.3. Objetivos

Cuando se va a realizar un trabajo como este es fundamental establecer unos objetivos claros para poder valorar de una manera mensurable la consecución de ellos. En este caso, establecer unos objetivos no es complicado, ya que el trabajo tiene un fin claro, pero sí que es cierto que se pueden establecer varios puntos que miden el avance del trabajo realizado. Los objetivos principales podrían resumirse en los siguientes puntos:

- **Comprensión de sistemas de aprendizaje profundo.** Cuando se comenzó con el desarrollo del trabajo se partía de un punto en el que se desconocía el funcionamiento y la idea que subyace detrás de los sistemas de aprendizaje profundo. Por lo tanto, el primer objetivo es tratar de comprender la idea general de estos sistemas, y en particular ser capaces de utilizar y explicar de forma detallada los empleados para la solución descrita. Es por esto que una parte de este texto recoge una explicación de estos sistemas.
- **Obtención de señal con información sobre eventos.** Realmente, una vez comprendidos teóricamente los sistemas de aprendizaje profundo, el siguiente paso es

ser capaces de aplicarlos para este caso particular y, dada una señal procedente del receptor, ser capaces de dar como respuesta otra señal que será el error de predicción y que tendrá valores altos en puntos de la fibra e instantes de tiempo en los que se ha producido un evento, mientras que para situaciones de normalidad tendrá valores inferiores. La consecución de esto supone facilitar la tarea de detección de eventos a un sistema de decisión final.

- **Sistema de decisión.** Esto supone dar una salida binaria al sistema global en términos de si hay un evento o no. Este no es un objetivo principal del trabajo, sino que pretende más bien ayudar a ejemplificar cómo partiendo de una señal procedente de un sistema DAS se puede obtener una señal que indique la presencia o ausencia de eventos.

De esta manera, comienza la explicación del método empleado para dicha tarea, incluyendo detalles sobre las señales entregadas por el receptor acústico y cómo se han tratado dichas señales para intentar sacar un máximo partido de ellas. Para ello, se hará también una breve revisión de los sistemas de aprendizaje profundo que mayoritariamente han servido para abordar el problema, así como posibles mejoras o alternativas sobre las que no se ha investigado por estar fuera del alcance temporal de este trabajo.

2. MARCO TEÓRICO

El objetivo de este capítulo es proporcionar al lector unas herramientas necesarias y suficientes para la comprensión del trabajo que se ha realizado.

2.1. Sensado acústico distribuido (DAS)

Aunque este no es un trabajo centrado en el campo de la óptica y su objetivo es más bien procesar y tratar de sacar el máximo partido de información proporcionada por sistemas que aprovechan características de la luz dentro de una fibra, vamos a hacer un repaso de los fenómenos físicos que permiten desarrollar sistemas como este.

La dispersión Rayleigh es la dispersión de una onda electromagnética por partículas de tamaño mucho menor a la longitud de onda de la radiación [7]. Aunque este es un fenómeno no deseable en la transmisión de información a través de una fibra óptica, puede tener aplicaciones, como es el caso [8].

Por otro lado, la reflectometría óptica coherente en dominio temporal (C-OTDR) es una técnica de medición utilizada para determinar algunas características de un medio de transmisión, como la fibra óptica, mediante la observación de las ondas reflejadas [9, 10]. Esto lo realiza utilizando la dispersión Rayleigh, que permite detectar señales de frecuencias acústicas a través de grandes distancias.

Se han presentado en los últimos años sistemas basados en C-OTDR que utilizan la fibra óptica como medio sensor para detectar vibraciones de forma distribuida (DAS) en el entorno físico próximo a la fibra. Éstos tienen la capacidad de detectar eventos con una muy buena precisión y habitualmente se han considerado útiles para sistemas de vigilancia, aunque, como hemos comentado ya, las aplicaciones de éstos pueden ir más allá.

En cuanto a la implementación de estos sistemas, habitualmente se emplea un láser coherente, cuya señal es amplificada e inyectada en una fibra óptica. La potencia óptica retroesparcida por la fibra sigue un patrón de interferencias por la suma coherente de los M frentes de onda generados por difusión Rayleigh en cada punto k de la fibra en función de sus fases Ω_i y amplitudes A_i según la expresión 2.1 [11].

$$R_k e^{j\theta_k} = \sum_{i=1}^{M} A_i e^{j\Omega_i} \tag{2.1}$$

Si la retrodifusión se mantiene constante en fase, la situación de interferencia producida por la suma coherente de la señal retroesparcida permanece constante. Sin embargo, la modificación de uno o varios elementos por una perturbación de tipo mecánica se traduce

en un cambio de fase de la señal retroesparcida en ese punto concreto, lo que da lugar a una variación en la situación de interferencia. Debido a este fenómeno la fibra se convierte en un sensor distribuido que permite detectar perturbaciones mecánicas en una posición concreta de la fibra.

Habitualmente, en cuanto al rango de funcionamiento de estos sistemas, se habla de unos 40-50 km, aunque esto depende evidentemente de la configuración óptica y de cuál es el evento que produce la perturbación mecánica. El porqué de esta limitación en distancia es simplemente que, por ejemplo, para una fibra monomodo y una longitud de onda de operación de 1550 nm se tiene una atenuación típica de 0.2 dB/km, por lo que para una longitud de fibra de 50 km en la que la luz ha de realizar un recorrido de ida y vuelta supone una atenuación de la señal de 10 dB y esto se traduce en un empeoramiento de la relación señal a ruido. En nuestro escenario de trabajo la longitud de la fibra es de 38.4 km, lo que supone que estamos cerca del límite que habitualmente se considera para estos sistemas y veremos como, efectivamente, cuanto mayor es la distancia a la que nos encontramos del receptor acústico, más complicado se vuelve detectar un evento.

Aquí es donde entra el sistema desarrollado en este trabajo, que supone una novedad en cuanto a la forma de tratar de encontrar estas perturbaciones mecánicas, y esta basado en conceptos propios del campo de la detección anomalías, es decir, se ha considerado que estas perturbaciones mecánicas son eventos anómalos que se pretende detectar.

2.2. Redes neuronales

Las redes neuronales artificiales (ANN) son modelos computacionales que tratan de imitar en cierta medida los procesos de aprendizaje y resolución de problemas de los humanos dando lugar a sistemas capaces de formarse a sí mismos en lugar de ser programados sobresaliendo, por tanto, en problemas que la programación convencional no es capaz de resolver de forma sencilla.

Aunque el surgimiento de la multitud de aplicaciones para las que las redes neuronales son útiles es algo ocurrido a lo largo de las últimas dos décadas, la realidad es que muchos de los algoritmos que se emplean hoy en día vieron la luz mucho antes, siendo en 1943 cuando surge el primer concepto que podemos relacionar con las actuales ANN [12].

2.2.1. La neurona

Primero, al estar una red neuronal formada por un conjunto de neuronas, esta es la parte por la que se ha de empezar esta explicación. El concepto detrás de esto no es complicado desde un punto de vista matemático ya que una neurona no es más que una función $T : \mathbb{R}^n \to \mathbb{R}$ no lineal tal y como se describe en 2.2. Para ello cada neurona tiene lo que se llaman pesos y sesgo, que se corresponde con las señales $\mathbf{w}$ y b, respectivamente, y que se van actualizando en el entrenamiento de la red, como explicaremos más adelante.

El tamaño de $\mathbf{w}$ es n, que es la dimensión de la entrada a la neurona y b es un escalar.

$$T(x) = f\left(b + \sum_{i=1}^{n} \mathbf{w}[i]x[i]\right) = f(b + \mathbf{w}^T x) \tag{2.2}$$

Donde $f : \mathbb{R} \to \mathbb{R}$ es una función no lineal que se conoce como función de activación y que es fundamental para el funcionamiento de una red neuronal, ya que es la parte que aporta la no linearidad al comportamiento de cada neurona y que permite modelar la realidad, que es en su mayoría no lineal.

2.2.2. La función de activación

En esta función es donde reside la esencia de las redes neuronales, ya que sin ella el comportamiento de la red sería lineal, algo que no tiene demasiado atractivo para modelar la inmensa mayoría de comportamientos. Aquí nos vamos a centrar en explicar brevemente qué característica fundamental se le ha de exigir a una función de activación..

El nombre nos da una idea de qué es lo que debe de hacer una función de activación y es que la idea fundamental y de la que se parte es que la neurona esté activada en unos casos y desactivada en otros. En cuanto a los casos en los que debe de estar activada se suele considerar que $f(\mathbf{w}^T x + b) > 0$. Esto subyace en la idea del comportamiento del entramado neuronal de los humanos y es que para ciertas acciones algunas neuronas se activan mientras que para otras se desactivan. El concepto de activación y desactivación nos podría llevar a pensar en la función signo, pero ésta no es utilizada en la práctica debido principalmente a su no derivabilidad en cero. En cambio, se utilizan habitualmente funciones que sí tienen como salidas a la mayoría de entradas valores muy concentrados en dos extremos que se corresponden con el concepto de activación y desactivación. En el anexo A se presentan las funciones de activación utilizadas en este trabajo y que son habitualmente empleadas.

2.2.3. De la neurona a la red neuronal - Forward Propagation

La idea de la neurona es interesante, pero al final es simplemente una composición de dos funciones, una lineal y otra que no lo es y que toman como entrada un vector. El poder de estos sistemas viene de la combinación de un número bastante alto de estas unidades. Para comprender esto mejor es común utilizar un diagrama como el que aparece en la figura 2.1. Cada uno de los conjuntos de neuronas que tienen como entrada un mismo vector se llama capa. Además, la expresión matemática que expresa la salida de una capa con respecto a su entrada es análoga a la de una única neurona. Más aún, la expresión a la salida de una capa k con n_k neuronas sigue manteniendo esta analogía cuando se tiene un conjunto de m entradas de longitud n_{k-1}, que coincide con el número de neuronas de la capa $k-1$ para $k > 1$ y con la longitud del vector de entrada a la red si $k = 1$. Esta

expresión se corresponde con la 2.3.

$$T(X_k) = F_k\left(B_k + W_k^T X_k\right) \tag{2.3}$$

donde:

- $W_k \in \mathbb{R}^{n_{k-1} \times n_k} := (w_{ij}^k)$ con $w_{ij}^k = \mathbf{w_j^k}[i]$ que contiene el peso i de la neurona j
- $X_k \in \mathbb{R}^{n_{k-1} \times m} := (x_{ij}^k)$ con $x_{ij}^k = \mathbf{x_j^k}[i]$ que contiene la posición i de la entrada j
- $B_k \in \mathbb{R}^{n_k \times m} := (b_{ij}^k)$ con $b_{ij}^k = \mathbf{b^k}[i] \forall\, j$ que contiene la el sesgo de la neurona i
- $F_k : \mathbb{R}^{n_k \times m} \to \mathbb{R}^{n_k \times m}$ dada por $F_k((a_{ij})) = (f_k(a_{ij}))$ donde $f_k : \mathbb{R} \to \mathbb{R}$ es la función de activación de la capa k

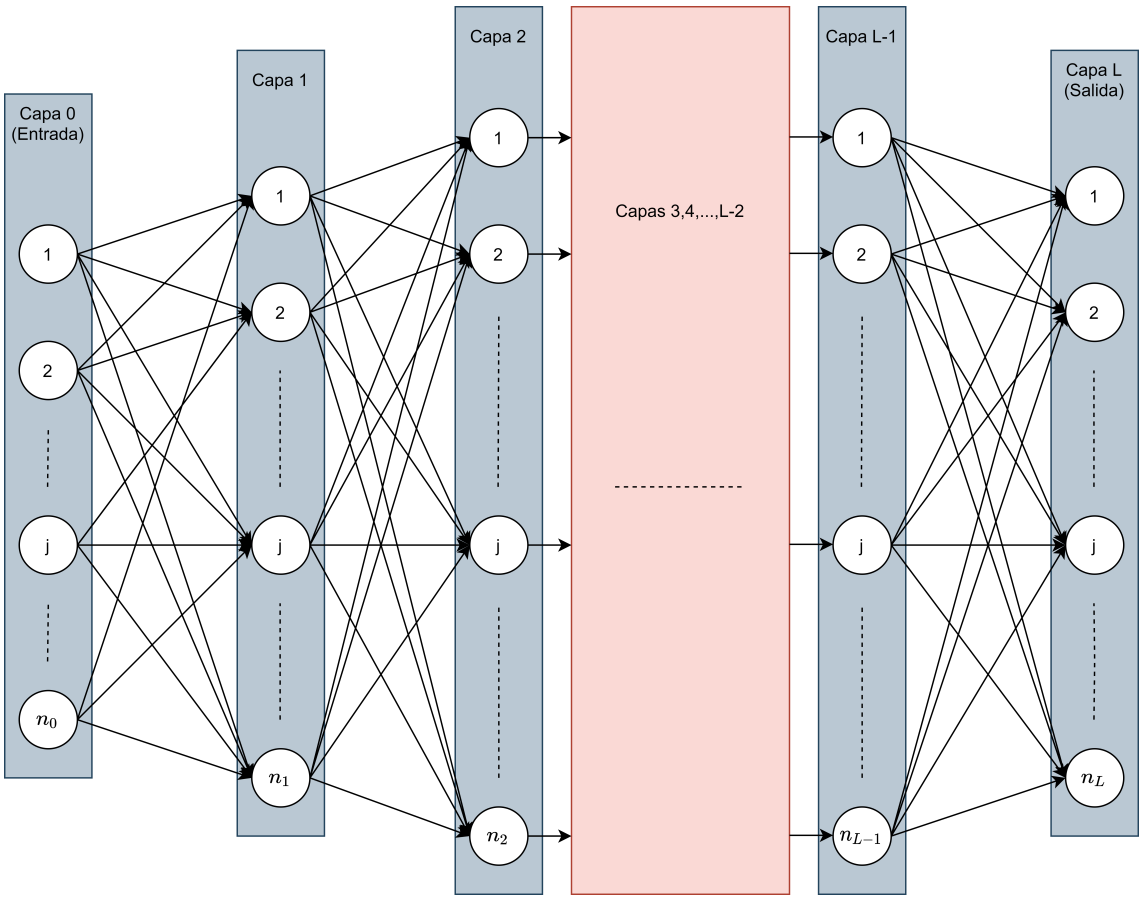


Fig. 2.1. Diagrama de una red neuronal con L capas

De aquí podemos ver varias cosas. Primero es que para obtener la salida de una capa de neuronas dado un conjunto de entradas simplemente se ha de realizar un producto y una suma de dos matrices para posteriormente evaluar cada uno de los elementos de la matriz resultante en una función de activación, la cual, según la expresión dada, es la misma para cada una de las salidas de cada uno de las neuronas de la misma capa. Esta praxis es común, aunque, como se puede ver fácilmente, no tendría por qué ser así.

Además, se ha comentado que la salida de la capa k con n_k neuronas tiene una entrada de longitud n_{k-1}, que es el número de neuronas de la capa anterior. Esto es así, porque la salida de la neurona $k-1$, que es de longitud n_{k-1}, sirve de entrada para la capa k. Es precisamente lo que se muestra en la figura 2.1 y es el funcionamiento básico de las redes neuronales, la conexión de una capa con la siguiente.

También, surge la pregunta de cómo elegir los valores de w_{ij} y b_{ij}. Pues bien, estos parámetros van cambiando en la etapa de entrenamiento, de hecho entrenar una red neuronal realmente significa buscar los valores de w_{ij} y b_{ij} que optimizan la acción a realizar. En cuanto a su inicialización suele ser con valores aleatorios cercanos a cero aunque esto es un campo de estudio muy amplio y se han propuesto diferentes técnicas [13, 14]. La forma en que se actualizan estos parámetros se explica a continuación.

2.2.4. Función de coste - Error cuadrático medio

Para medir el rendimiento de una red neuronal se deben emplear métricas que reflejen cómo de bueno es. Para ello se definen las funciones de coste, que de alguna forma relacionan la salida obtenida por la red (y) y la salida objetivo (y_t). Se suelen emplear distintas funciones de coste atendiendo a si el tipo de tarea que realiza la red es de regresión o de clasificación. La regresión, que es a la que pertenece este sistema, se puede resumir como el procedimiento de predecir valores, mientras que la tarea de clasificación consiste en elegir de entre varias una clase a la que pertenece la entrada.

Entre las funciones de coste para regresión, se emplea habitualmente el error cuadrático medio (MSE) y es la única que vamos a explicar aquí por ser la que emplearemos más adelante. También cabe recalcar que, a pesar de que el error cuadrático medio es una métrica muy conocida por emplearse para la optimización de redes neuronales, sus usos no se reducen exclusivamente a ello y, de hecho, lo utilizaremos también en otro ámbito más adelante. En la expresión 2.4 se define el error cuadrático para dos vectores cualesquiera $\mathbf{y}$ e $\mathbf{y_t}$ de longitud N. También se puede definir esta medida de error para señales n-dimensionales. Para este caso en el que vamos a trabajar con imágenes, conviene definirlo concretamente para dos matrices Y e Y_t de tamaño $N \times M$, como se muestra en la expresión 2.5.

$$MSE(\mathbf{y}, \mathbf{y_t}) = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{y}[n] - \mathbf{y_t}[n])^2 \tag{2.4}$$

$$MSE(Y, Y_t) = \frac{1}{MN}\sum_{n=1}^{M}\sum_{m=1}^{N}(Y[m,n] - Y_t[m,n])^2 \tag{2.5}$$

Aprovechamos para describir aquí una métrica de error que, aunque no se emplea como función de coste, ya que no es una función $f : \mathbb{R}^{N\times M} \times \mathbb{R}^{N\times M} \rightarrow \mathbb{R}$, pero que nosotros emplearemos en algún momento para comparar dos señales bidimensionales y que es el error cuadrático instantáneo y se describe según la expresión 2.6.

$$ISE(Y, Y_t) = (Y - Y_t)^2 \tag{2.6}$$

Para casos en los que la tarea de la red neuronal es la clasificación, se emplean otras métricas que aquí no vamos a describir porque no las vamos a necesitar para desarrollar el trabajo y que están basadas en conceptos de teoría de la información. La más utilizada es la entropía cruzada, que parte del concepto de entropía que se describió por primera vez en 1948 [15].

Se ve inmediatamente que estas medidas de error dependen de los valores w_{ij}^k y b_{ij}^k, ya que la salida de la red también depende de éstos. La fase de entrenamiento de una red consiste entonces en tratar de encontrar los valores de cada w_{ij}^k y b_{ij}^k que optimizan la salida de la función de coste. Así pues, éstas son funciones multivariable cuyas variables son la salida de la red Y, la salida objetivo Y_t, los pesos w_{ij}^k y los sesgos b_{ij}^k. Por lo tanto, el entrenamiento para el caso del MSE se puede ver como la búsqueda de mínimos de una función multivariable que depende de los pesos y sesgos de cada una de las neuronas en la red, así como de la entrada y la salida objetivo, $J(Y, Y_t, \theta)$, donde θ es un objeto que contiene todos los pesos y sesgos. La explicación procedimental aparece a continuación.

2.2.5. Actualización de los parámetros - Backward Propagation y Adam

Como acabamos de explicar, el proceso de entrenamiento de una red consiste idealmente en encontrar el mínimo global de una función $f : \mathbb{R}^n \to \mathbb{R}$. Habitualmente, cuando se trata de buscar mínimos de funciones multivariables, se recurre al concepto de gradiente, pero cuando n es muy grande esto no es para nada trivial. Este es el caso de una red neuronal, ya que hemos de recordar que las variables son Y, Y_t, w_{ij}^k y b_{ij}^k, lo que en general da un problema complicado.

Para solucionar este problema, se han desarrollado distintos algoritmos en los que se suele valorar la velocidad a la que realizan la tarea de encontrar mínimos (convergencia) y lo eficaces que son en ella. En 1847 ya se propuso una primera solución que aún a día de hoy sigue en uso salvo algunas modificaciones [16]. En muchos casos, como el nuestro, conviene penalizar si es necesario el funcionamiento por un tiempo de convergencia bajo, es decir, por encontrar rápidamente un mínimo. Para estos casos, es ampliamente empleado un algoritmo llamado Adam. Su nombre deriva de adaptive moment estimation, lo que nos da una idea de en qué se basa su funcionamiento. Aquí vamos a dar las ecuaciones de actualización de los pesos, ya que es suficiente para dar una idea básica de cuál es el objetivo del algoritmo y su funcionamiento. No obstante, en [17] se puede encontrar el desarrollo del algoritmo más en detalle.

El problema de tratar este asunto desde un punto de vista iterativo en lugar de analítico es que el objetivo ideal es encontrar el mínimo global de la función de coste, pero es algo común que una vez se encuentra un mínimo local, el algoritmo considere que es global y no llegue nunca a este último. Esto se entiende muy bien en la figura 2.2 en la que se muestra un caso trivial de una función de coste que depende una única variable. El punto verde se corresponde con un mínimo local, pero que no es global a pesar de que un algoritmo iterativo podría considerarlo como tal y nunca se llegaría a alcanzar el punto

rojo, que es realmente el mínimo global de la función.

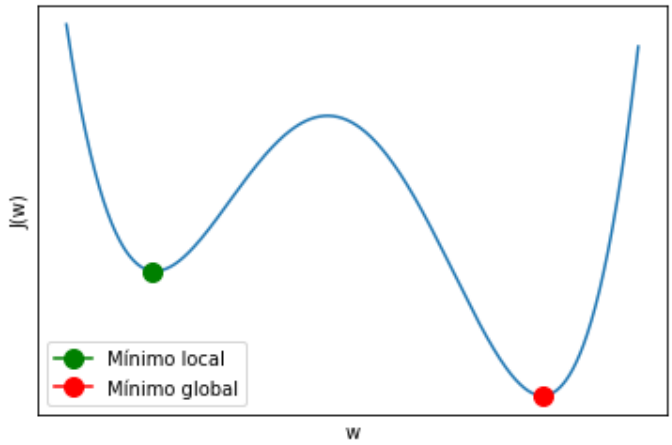


Fig. 2.2. Ejemplo de función de coste de una variable con un mínimo local y un mínimo global

Para explicar este algoritmo vamos a renunciar por un momento a la notación empleada hasta aquí, por simplificar un poco las expresiones y vamos a considerar simplemente una función de coste que depende de dos señales x_t e y_t y un objeto θ. Partimos entonces de la base de que vamos a describir un algoritmo iterativo que permite buscar mínimos de una función de coste multivariable $J(x_t, y_t, \theta_{t-1})$ de forma que x_t e y_t son la entrada y salida de la iteración t, respectivamente y θ_{t-1} contiene todos los coeficientes w_{ij}^k y b_{ij}^k de la red obtenidos en la iteración $t-1$ del entrenamiento para $t > 1$. Por lo tanto, el objetivo es ver cómo se actualizan el valor de θ, esto es, cómo se pasa de θ_{t-1} a θ_t. Este proceso se muestra a continuación. Para ello, se ha de contar primero con el valor de los pesos y sesgos en la primera iteración θ_0 que, como hemos comentado, suelen ser valores aleatorios pequeños. También toma cuatro parámetros de entrada, que son β_1, β_2, α (learning rate) y ϵ.

$$
\begin{aligned}
m_0 = v_0 &= 0 \\
\nabla_t &= \frac{\partial J(x_t, y_t, \theta_{t-1})}{\partial \theta_{t-1}} \\
m_t &= \beta_1 m_{t-1} + (1-\beta_1)\nabla_t = (1-\beta_1)\sum_{i=1}^{t} \beta_1^{t-i}\nabla_i \\
v_t &= \beta_2 m_{t-1} + (1-\beta_2)\nabla_t = (1-\beta_2)\sum_{i=1}^{t} \beta_2^{t-i}\nabla_i \\
\hat{m}_t &= \frac{m_t}{1-\beta_1^t} \\
\hat{v}_t &= \frac{v_t}{1-\beta_2^t} \\
\theta_t &= \theta_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}
\end{aligned}
\tag{2.7}
$$

Surge la duda aquí de, dado un conjunto X e Y de datos para el entrenamiento, cómo escoger x_t e y_t, es decir, qué datos utilizar para la actualización de θ en la iteración t. Aquí

aparece un concepto nuevo y es el de batch (lote). Esto hace referencia a que una forma eficiente de entrenar la red es elegir lotes de datos de longitud habitualmente fija que se llama tamaño de batch (B). Esto significa que x_t e y_t son vectores de longitud fija B.

Además, es común que, dado este conjunto X e Y de datos, se utilicen varias veces para el proceso del entrenamiento. El motivo puede verse simplemente como que utilizar los datos sólo una vez es insuficiente para la convergencia del algoritmo. Surge aquí el concepto de epochs (épocas), que se refiere a que la epoch n es la n-ésima vez que se utiliza un lote de datos para entrenar la red. El número de veces que se emplean los datos se conoce como número de epochs.

2.2.6. Normalización de las entradas

Un tema de estudio de las redes neuronales es el de cómo normalizar las entradas para permitir al algoritmo de entrenamiento que acabamos de describir encontrar los mínimos. El valor del gradiente de una función depende evidentemente de la magnitud de las entradas de dicha función. En este caso, $\nabla_t = \frac{\partial J(x_t, y_t, \theta_{t-1})}{\partial \theta_{t-1}}$ depende de la magnitud de x_t e y_t. Por lo tanto, el valor de éstos es fundamental para una convergencia eficiente del algoritmo, lo que hace de esto una parte sencilla, pero fundamental para el correcto funcionamiento de las redes neuronales. Dado que normalizar es un concepto poco concreto, aparecen aquí dos términos, que son la estandarización y escalado [18].

Primero, en cuanto a la estandarización, consiste en manipular los datos de forma que sigan una distribución de probabilidad concreta, habitualmente con media cero y varianza uno. Este concepto no se ha empleado en este trabajo, ya que podría afectar a las estadísticas de los datos anómalos, cosa que desde luego no nos interesa.

Por otro lado, el escalado, consiste en convertir todos los datos a un rango de valores como, por ejemplo, de cero a uno, que es el que se ha empleado en este trabajo y que se llama escalado MinMax, que es simplemente que dado un conjunto de datos x la forma en la que se obtienen los datos escalados y responde a la expresión:

$$y = \frac{x - min(x)}{max(x) - min(x)} \tag{2.8}$$

2.2.7. Capas convolucionales

La explicada no es la única forma de conectar las capas adyacentes de una red neuronal. De hecho, en este trabajo se han empleado lo que se conocen como capas convolucionales, que emplean la correlación cruzada para conectar las capas de la red. En el anexo B aparece un desarrollo matemático de este tipo de capas.

2.3. Autoencoders

Un autoencoder es un tipo especial de red neuronal que tiene dos partes principales: el codificador y el decodificador, y cuyo objetivo es que la entrada del codificador y la salida del decodificador sean lo más parecidas posible. Como cabe esperar, la salida del codificador es la entrada del decodificador y, aunque, en general, no se pueden modelar estos elementos como funciones matemáticas invertibles, podríamos asemejarlo al concepto de funciones inversas. La señal que está a la salida del codificador se llama código y es habitualmente de menor tamaño que la señal de entrada. Esto se entiende bien en la figura 2.3.

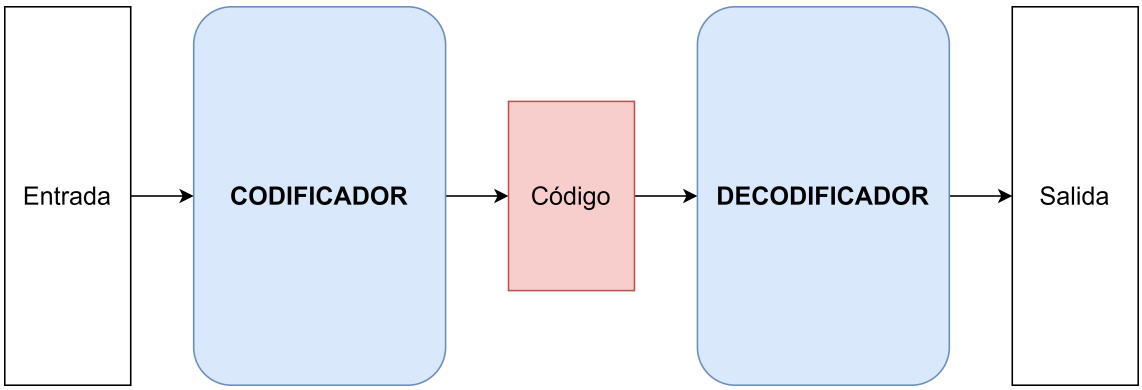


Fig. 2.3. Esquema básico de un Autoencoder

El concepto de codificar y decodificar que se acaba de explicar es empleado en multitud de aplicaciones informáticas cuando interesa disminuir el tamaño que un fichero ocupa en memoria, o cuando se pretende que la transmisión de información entre dispositivos sea lo más eficiente posible. Tradicionalmente, se han investigado formas de codificar la información de manera que se extrajeran las características más relevantes del tipo de archivo en cuestión. Habitualmente, en este proceso de codificación y decodificación se admiten pérdidas, esto es, la entrada al codificador y la salida del decodificador no son exactamente iguales, pero sí que conservan las propiedades más importantes del fichero. Un ejemplo de esto es el formato de imagen JPEG [19], el cual se aprovecha de la poca resolución que tiene el ojo humano para cambios bruscos de texturas y color para reducir el tamaño de las imágenes. En la figura 2.4 se comparan una misma imagen en dos formatos, PNG sin pérdidas y JPEG con pérdidas y el tamaño que ocupan cada uno de los ficheros. Claramente, a pesar de haber reducido su tamaño a menos de la cuarta parte, debemos de fijarnos con mucho detalle para percibir las diferencias.

Un autoencoder, por lo tanto, realiza el tradicional proceso de codificar y decodificar, pero su punto fuerte reside en que no es necesario que un humano le dicte cuál es el proceso a realizar sino que, mediante el procedimiento tradicional que se emplea para entrenar a redes neuronales, la propia red es capaz de extraer las características más relevantes de las señales con las que habitualmente se va a trabajar. Así pues, una red con esta estructura se suele ver como una red capaz de extraer características relevantes de un conjunto

(a) Imagen PNG (462 KB) (b) Imagen JPEG (104 KB)

Fig. 2.4. Una misma imagen en formato PNG (sin pérdidas) y JPEG (con pérdidas)

de datos por sí sola, lo cual es fundamental en infinidad de aplicaciones como la que nos atañe. De hecho, y muy en relación a lo anterior, se han utilizado para comprensión con pérdidas en imágenes dando unos resultados notables y comparables con el formato JPEG [20].

2.4. Detección de anomalías y autoencoders

La detección de anomalías es un campo ampliamente estudiado y para el que se han desarrollado un gran número de métodos para conseguir objetivos relacionados con la detección y clasificación de eventos anómalos [21, 22]. Habitualmente, se hace una división entre detección de anomalías mediante un método supervisado y uno no supervisado. Además, no es, en general, un problema con solución binaria en el que un suceso es anómalo o no, sino que el objetivo va más allá y se pretende detectar anomalías para posteriormente clasificarlas, en caso de existir distintas categorías.

El primer caso es útil y eficaz cuando se tienen los datos etiquetados, ya que el método consiste básicamente en ser capaces de encontrar qué función de densidad de probabilidad describen algunos parámetros en cada uno de los posibles sucesos (normales y todos los tipos de anomalías) y, con esto y haciendo uso de conceptos de la teoría de la decisión, ser capaces de determinar a qué categoría pertenece cada suceso. Este caso, aunque se han desarrollado formas de tratarlo desde un punto de vista teórico, es poco común en escenarios de aplicación, ya que las anomalías suelen tener una frecuencia de aparición muy baja y el modelado de sus características puede ser una tarea irrealizable.

Por otro lado, la detección de anomalías para el caso no supervisado trae consigo una serie de dificultades añadidas con respecto al caso anterior. Primero, no se tienen los

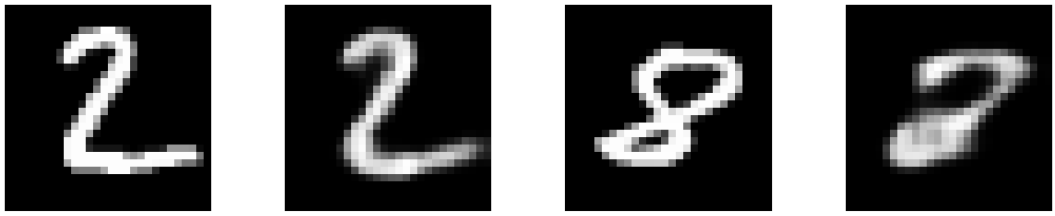
datos etiquetados, lo que significa que no se tiene una información a priori de si un suceso concreto es anómalo o no. Además de esto, no se pueden extraer características de un tipo de suceso, pues no se sabe cuáles de los datos se corresponden con ese tipo de sucesos.

Para este último caso, el cual es más relevante para este trabajo, se han propuesto soluciones alternativas basadas en algoritmos, algunos complejos y con un rendimiento no del todo bueno para según qué entornos. Un tipo de algoritmo que describen una solución interesante y relativamente sencilla de implementar y comprender son los basados en agrupamiento o clustering y consisten en agrupar los datos en un número K de grupos, que es un parámetro de entrada del algoritmo, y a través de métricas basadas en el concepto matemático de distancia entre dos vectores, es capaz de segmentar todos los datos en estos K grupos [23, 24]. Este tipo de algoritmos es especialmente interesante cuando se tiene un número fijo de posibles situaciones anómalas, aunque se han presentado variables en las que, partiendo de un desconocimiento de cuál es el número de grupos, son capaces de estimarlo [25].

Como alternativa a los métodos basados en algoritmos, surgen ideas basadas en el uso de redes neuronales, en particular de autoencoders [26, 27]. La idea fundamental que hay detrás de todo esto es sencilla e intuitiva. Como se ha explicado en la sección anterior, el objetivo es que la entrada y salida de éstos sea lo más parecida posible tras haber pasado por una versión comprimida de los datos. Para esto, las redes se entrenan con unos datos para que extraigan las características más relevantes de éstos. Así, si tras haber entrenado la red para codificar y decodificar un tipo de información en concreto, trata de codificar otro tipo de datos el rendimiento será claramente peor. Esto último es precisamente el principio en el que se basa el empleo de autoencoders como método de detección de anomalías, en entrenar las redes con datos normales de manera que éstos los reconstruya con un error mínimo, pero para los datos anómalos, en cambio, obtenga un error de una magnitud superior. Así pues, este método es ideal para casos en los que se busca una salida binaria que diga si un elemento del conjunto de datos es normal o anómalo.

Para ayudar a explicar esto, se ha creado un autoencoder sencillo y se ha hecho uso de la base de datos MNIST [28]. Este autoencoder ha sido entrenado para codificar y decodificar imágenes que contienen un número dos manuscrito de distintas maneras. Posteriormente, se le han pasado como entrada dos imágenes, una que contiene un dos y otra que con un ocho, que haría las veces de anomalía. En la figura 2.5 se muestran los resultados. Se ve claramente cómo la imagen que contiene el número dos ha sido bien reconstruida, mientras que la que contiene el número ocho no lo ha sido tanto e incluso puede dar la impresión de que ha reconstruido bien la parte común que tienen el dos y el ocho.

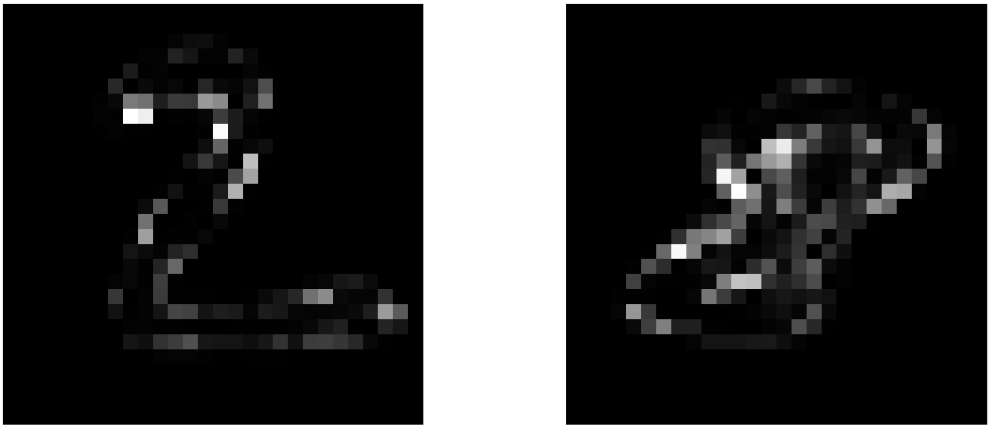
De hecho, en la figura 2.6 se muestra el error cuadrático instantáneo para la reconstrucción del dos y el ocho. Se ve que para el caso del dos las zonas más problemática son los bordes, pero que en la parte correspondiente al interior del número el error es mínimo. En cambio, en el ocho los bordes tienen un error alto, pero es que además en alguna zona



(a) 2 original (b) 2 decodificado (c) 8 original (d) 8 decodificado

Fig. 2.5. Comparación reconstrucción de un número 2 y un número 8 con un autoencoder entrenado para reconstruir imágenes con números 2

de la parte interior del número el error sigue siendo alto y, de hecho, en las zonas que es bajo es por su similitud con el número 2. Además, si calculamos el error cuadrático medio para ambos casos, para el dos el valor es 0.0141, mientras que para el ocho es 0.0348, hecho que permitiría detectar el ocho como una anomalía fácilmente.



(a) ISE para el número 2 (b) ISE para el número 8

Fig. 2.6. Error cuadrático instantáneo de la reconstrucción de un número 2 y un número 8 con un autoencoder entrenado para reconstruir imágenes con números 2

2.5. Enventanado de señales bidimensionales

Una técnica bastante utilizada en el procesado de señales y que será empleada a lo largo de este trabajo en varias ocasiones es el enventanado. Su idea es sencilla, pero conviene definir algunos parámetros. Además, aunque el principio de su funcionamiento resulta intuitivo para señales unidimensionales, no lo puede resultar tanto para casos con más dimensiones, por lo que se plantea una explicación que aclara qué es lo que entendemos aquí por un enventanado en una y dos dimensiones.

Empecemos definiendo qué es el enventanado de señales unidimensionales. Hay dos factores que llamaremos N y M que hacen referencia a la longitud y desplazamiento de ventana, respectivamente. Dados estos factores y una señal unidimensional x de entrada con longitud L, el primer paso es obtener una matriz cuyas filas contienen cada una de las partes en los que se divide dicha señal, a las que llamaremos ventanas. Esto se define en la expresión 2.9. Posteriormente, cada una de estas filas se multiplica punto a punto con lo que se conoce como una función ventana. En el procesado de señales se emplean múltiples funciones ventanas [29], pero la más sencilla y la que vamos a emplear en esta ocasión es la rectangular, esto es, la función constante con amplitud 1, lo que supone dejar la matriz obtenida tal y como está.

$$\begin{gathered} x \in \mathbb{R}^L \text{ de forma que } x = (x[1], x[2], ..., x[L]) \\ f : \mathbb{R}^L \to \mathbb{R}^{WxN} \text{ donde W} = \left\lfloor \frac{L-N}{M} + 1 \right\rfloor \text{ y es el número de ventanas} \\ \text{De forma que } f(x) = (a_{ij}) \text{ donde } a_{ij} = x[M(i-1)+j] \end{gathered} \tag{2.9}$$

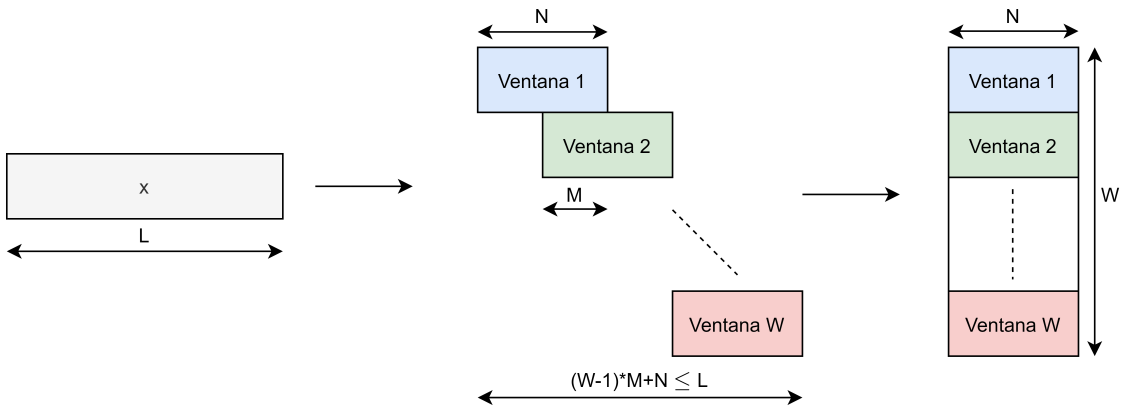


Fig. 2.7. Proceso de enventanado para función de ventana rectangular

Al extrapolar este concepto a señales bidimensionales se considera que la señal de entrada es de tamaño $L_1 \times L_2$. Se han de definir además cuatro parámetros, que son N_i y M_i $(i = 1, 2)$ y que se refieren a la longitud y el desplazamiento de la dimensión i, respectivamente y las ventanas por lo tanto son matrices por lo que nos queda a la salida una matriz de matrices, o una señal de cuatro dimensiones. Esta matriz de matrices tiene un tamaño de $W_1 \times W_2$ por lo que la señal de cuatro dimensiones es de tamaño $W_1 \times W_2 \times N_1 \times N_2$. Por lo tanto, dada una señal bidimensional $x \in \mathbb{R}^{L_1 \times L_2}$ la función enventanado f se define por:

$$\begin{gathered} f : \mathbb{R}^{L_1 \times L_2} \to \mathbb{R}^{W_1 \times W_2 \times N_1 \times N_2} \text{ dada por } f(x) = (a_{i_1 i_2 j_1 j_2}) \text{ donde} \\ W_i = \left\lfloor \frac{L_i - N_i}{M_i} + 1 \right\rfloor \text{ y es el número de ventanas de la dimensión i y} \\ a_{i_1 i_2 j_1 j_2} = x[M_1(i_1-1)+j_1, M_2(i_2-1)+j_2] \end{gathered} \tag{2.10}$$

En este trabajo, lo que hemos realizado posteriormente es convertir esta matriz de matrices en un vector de matrices, algo que se conoce como flattening [30] ya que, como

veremos más adelante, nuestra unidad de trabajo es la ventana y en rasgos generales no es relevante su posición espacial y temporal. Esto último no es cierto por un pequeño detalle y es que utilizaremos distintas redes neuronales para distinas partes de la fibra, algo que detallaremos más en el siguiente capítulo. Además, también se podrían definir señales ventana bidimensionales, pero nosotros hemos empleado la función constante de amplitud de valor uno, es decir, hemos dejado la señal tal y como está.

3. MARCO METODOLÓGICO

Este capítulo es el tronco para comprender el funcionamiento final del sistema. Tras conocer y haber asimilado los conceptos explicados en el capítulo anterior, en éste se da una explicación de cómo cada una de las piezas teóricas conforman el entramado práctico que permite solucionar de forma eficaz el problema inicialmente propuesto. También se explican la elecciones de los aspectos clave para el correcto funcionamiento del trabajo, las cuales son basadas mayoritariamente en un criterio empírico. Es cierto también que, a pesar de explicarse estas elecciones, no se ahonda en exceso en muchas de ellas porque ello supondría presentar un número demasiado alto de comparaciones entre distintas posibilidades de parámetros de diseño, algo que resultaría en un informe excesivo y que carece de verdadera trascendencia para la comprensión del sistema y su interés desde un punto de vista teórico.

3.1. Captura de la información

Aunque la forma en que la información es capturada por el receptor acústico distribuido no es realmente parte de este trabajo, sí que conviene presentar el escenario en el que se tomaron las medidas con las que se ha trabajado, ya que esto nos permitirá principalmente familiarizarnos con las posiciones en las que potencialmente están los eventos. En la figura 3.1 se incluye un esquema que permite entender bien el escenario en el que se tomaron las medidas.

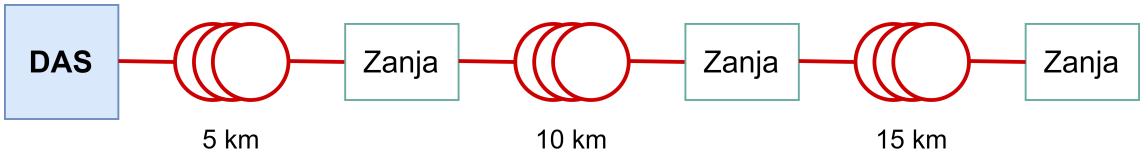


Fig. 3.1. Esquema del escenario de toma de medidas

Realmente, la longitud de fibra con las que se ha trabajado es de 38.4 km, pero su configuración es de bobinas de 5, 10 y 15 kilómetros enrolladas y una zanja, que es donde se han producido los eventos. Para hacer una idea del escenario físico donde se hicieron las pruebas, en la figura 3.2 se puede ver en el día en que se realizaron las pruebas.

Hay que adelantar aquí un aspecto y es que tal y como se muestra en la figura 3.1, el escenario donde se tomaron las medidas pretende simular uno final en el que hay una fibra de 38.4 km desplegada en el suelo. Este hecho genera un pequeño problema y es que según se ha podido comprobar el comportamiento de la luz dentro de la fibra no es el mismo en los carretes que en la zona de las zanjas. Por lo tanto, a la hora de que el autoencoder generalice su aprendizaje y teniendo en cuenta que la mayoría de las medidas con las

Fig. 3.2. Escenario físico donde se tomaron las medidas

que se ha trabajado son de zonas pertenecientes fibra en carretes, las zonas de zanja, simplemente por el hecho de estar en una zanja generarán un mayor error de predicción aún en casos en los que no hay eventos. Esto se verá un poco más en detalle más adelante en el capítulo 4

3.2. Preprocesado de la información

Antes de comenzar a explicar la solución que se ha dado al problema, es primordial conocer la situación de partida. Como se ha comentado en el capítulo 1, las señales desde las que se parte son bidimensionales, siendo una dimensión el tiempo y otra la distancia. Pero antes de eso, los datos proporcionados por el receptor no están en este formato, por lo que se ha de proceder a acondicionarlos y convertirlos precisamente a esta señal bidimensional. El proceso completo para convertir este fichero binario en la entrada a la red es lo que llamamos preprocesado y se va a explicar en esta sección.

3.2.1. Información capturada por el sensor

No nos vamos a detener apenas en explicar la conversión de la información procedente del DAS a la matriz, pero sí que es interesante comentar que aquí aparece un concepto utilizado en el ámbito de las redes convolucionales y es el de diezmado. La señal que nos entrega el sensor acústico tiene una resolución espacial de 6.4 metros y una resolución temporal de un milisegundo o una frecuencia de muestreo de 1kHz. Esto proporciona una cantidad de datos demasiado alta que puede ralentizar su procesado. Es por este motivo, que se aplica un factor de diezmado temporal $Dt = 10$, reduciendo así la frecuencia

de muestreo a 100Hz y teniendo una resolución de 10 milisegundos, que sigue siendo más que aceptable. El concepto de diezmado que ya se ha explicado es realmente una reducción de la frecuencia de muestreo cuando se realiza una conversión del espacio analógico al digital.

Cabe comentar que se probaron distintos factores de diezmado y se observaron varias cosas. La primera es que los resultados son mejores cuanto menor es el factor de diezmado, puesto que la cantidad de información con la que trabajar es mayor. Por otro lado, mayor información con la que trabajar supone un mayor tiempo de procesado y no hay que perder de vista que el fin último de este sistema es funcionar en tiempo real. Por este motivo, se probó a incrementar el factor de diezmado y aunque no se presentan resultados mensurables se vio que el sistema funcionaba aceptablemente incluso con $Dt = 50$, aunque evidentemente el rendimiento disminuía a cambio de disminuir el tiempo de procesado considerablemente. Todos los resultados aquí presentados son con $Dt = 10$, ya que supone un punto medio.

3.2.2. Formato de las señales bidimensionales

Como se ha explicado, la resolución de la dimensión temporal tras el diezmado es 10 milisegundos, lo que se traduce en que un minuto de información capturada tenemos 6000 muestras. Las señales con las que se ha trabajado para este trabajo tienen duraciones que son múltiplos de un minuto y, aunque no tiene por qué ser así y el sistema funcionaría perfectamente con otras duraciones, para explicar más adelante cómo se ha realiza el enventanado, supondremos que tenemos señales de un minuto (6000 muestras).

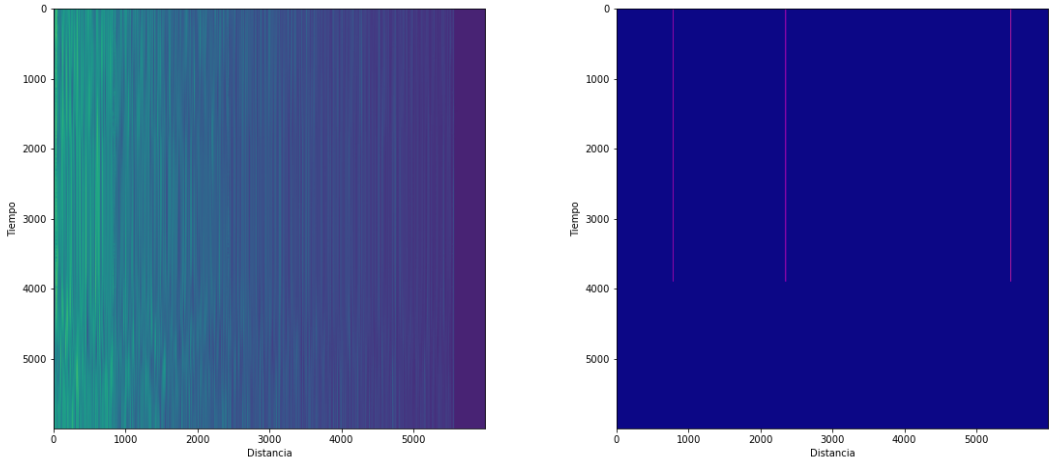
Por otro lado, la resolución espacial, como se ha explicado también, es de 6.4 metros. Esto, añadido a que las señales con las que se ha trabajado corresponden a un tramo de 38.4 kilómetros de fibra resulta en que también hay 6000 muestras en la dimensión espacial. Igualmente, el sistema está preparado para otras longitudes de fibra.

Así pues, con esto, para un tiempo de un minuto, lo cual consideraremos a partir de ahora nuestra unidad de trabajo salvo que se indique lo contrario, la señal sin más procesado que un diezmado temporal es una matriz de 6000×6000. Esta señal será enventanada y estas ventanas serán las que se utilicen como entrada a la red neuronal.

En la figura 3.3a se muestra un ejemplo de una captura de un minuto de duración y en el que existe evento en varias distancias y en varios instantes de tiempo que aparecen en rojo en 3.3b. En esta comparación es muy sencillo ver que a simple vista no es posible distinguir con claridad anomalías dada la señal de entrada.

3.2.3. Enventanado de la matriz

Esta sección aclara cuál es la entrada a la red neuronal, por lo que resulta clave para el entendimiento del sistema global y, aunque su comprensión no es compleja, hay ciertas



(a) Matriz 6000x6000 del receptor y diezmada (b) Matriz 6000x6000 con los eventos

Fig. 3.3. Comparación de las matrices entrada y salida esperada del sistema

decisiones tomadas durante la fase de diseño del sistema que quedan plasmadas aquí y resultan interesantes y de las que se pueden extraer conclusiones. Recalcar además que muchas de las decisiones tomadas de ahora en adelante son en base a un criterio empírico y a trabajos fundados en un método de prueba y error, ya que no hay apenas literatura que consultar sobre este tema.

Como se ha explicado en la sección 2.5, cuando se va a realizar un enventanado bidimensional se han de definir cuatro parámetros, a los cuales llamaremos N_t, M_t, N_s y M_s. Las letras N y M se emplean para la longitud y desplazamiento de ventana, respectivamente; y los subíndices t y s para las dimensiones temporal y espacial, respectivamente.

Primero, en cuanto a la dimensión espacial, se ha tomado que $N_s = M_s$, lo cual implica que el solape entre ventanas es nulo. Esto es así porque el solape espacial no tiene demasiado sentido aquí, ya que, a diferencia de en la dimensión temporal, no buscamos continuidad entre ventanas. Por otro lado, el valor de N_s se vio que proporcionaba unos mejores resultados cuanto menor era, pero ralentizaba el procesado de todos los datos debido a que supone un mayor número de ventanas, lógicamente. Es por esto, que se eligió un valor de $N_s = M_s = 8$, lo que proporciona un equilibrio entre rendimiento y tiempo.

Por otro lado, en cuanto a la dimensión temporal, sí que se ha empleado un valor de M_t distinto a N_t, algo que tiene como objetivo aportar mayor continuidad entre ventanas consecutivas temporalmente. En cuanto a qué valores se han empleado aquí, se observó que realmente el valor de N_t no influía demasiado en el rendimiento del sistema. Por lo tanto, se eligió 40, ya que es un tamaño intermedio que nos permite tener una mayor flexibilidad que con ventanas de mayor tamaño. Como contrapartida, utilizar un valor superior de N_t se reflejaría en una disminución del tiempo de procesado, lo que podría ser interesante en muchos casos. En cuanto al valor de M_t, se eligió 30, por proporcionar

cierto solape entre ventanas pero sin aumentar en exceso la información a procesar.

3.2.4. Agrupación de ventanas por distancia

Como cabía esperar, el comportamiento de la señal dentro de la fibra varía a lo largo de la misma ya que la luz sufre una atenuación que aumenta con la distancia, por lo que la potencia que llega a distancias mayores es menor y empeora así la relación señal a ruido. Esto supone que tratar todas las ventanas de la misma manera y de forma independiente a la distancia que corresponden es eficaz pero ineficiente. Una buena medida para solucionar esto sin añadir dificultad conceptual ni de procesamiento es emplear varios autoencoders, cada uno especializado en un segmento de la fibra. Concretamente, para este trabajo se vio que se obtenían buenos resultados utilizando cuatro autoencoders distintos, aunque como sucede con prácticamente cada una de las elecciones, puede ser cambiado para casos en los que, por ejemplo, la fibra es de mayor o menor longitud, o la señal se muestra más o menos ruidosa.

Las longitudes de cada una de estas cuatro divisiones no son iguales, pues la atenuación de la señal en la fibra se corresponde a un valor aproximado de 0.2 dB/km para una fibra de vidrio monomodo trabajando en 1550 nm. Este efecto se apreció perfectamente cuando se realizaron las primeras aproximaciones, en las que se empleó un único autoencoder, ya que los eventos ocurridos en distancias más cercanas al sensor eran detectados perfectamente, mientras que los más alejados resultaban más desapercibidos. Este efecto, aunque es inevitable, motivó la idea de emplear distintas redes para distintas partes de la fibra.

Además de esto, en los últimos 2.8 km medidos, debido a la configuración y situación física de la fibra, se obtienen datos completamente ruidosos y que no contienen ninguna información por lo que realmente la longitud efectiva de la fibra con la que se ha trabajado es de 35.6 km. Por todo lo comentado, las distancias que cubre cada uno de los cuatro autoencoders son las mostradas en la tabla. 3.1

	Distancias para las que ha sido empleado (km)
Autoencoder 1	0 - 17.92
Autoencoder 2	17.92 - 25.60
Autoencoder 3	25.60 - 30.72
Autoencoder 4	30.72 - 35.60

TABLA 3.1. DISTANCIAS CUBIERTAS POR CADA UNO DE LOS AUTOENCODERS

3.2.5. Esquema del preprocesado

Finalmente, se incluye un esquema en la figura 3.4 que resume cada una de las partes anteriormente explicadas atendiendo también a las dimensiones de las señales en cada una de las fases. Como ya se ha apuntado, los tamaños de las señales son para mediciones de un minuto.

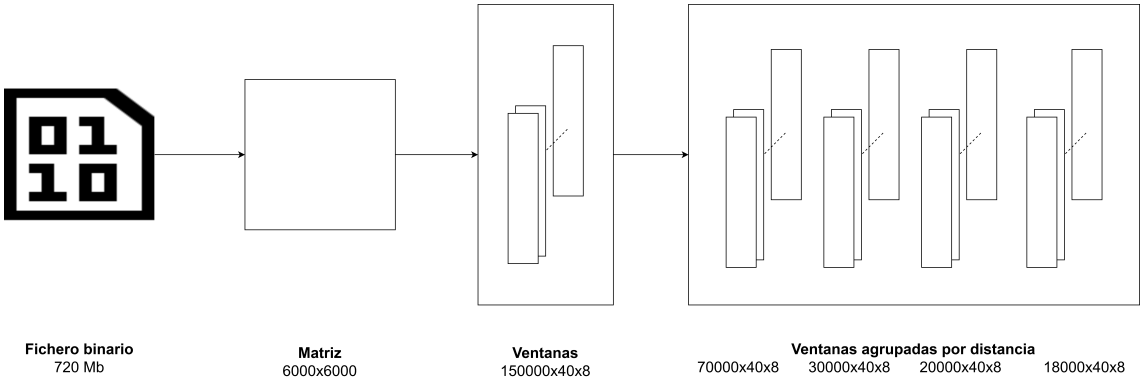


Fig. 3.4. Esquema del preprocesado

3.3. Procesado de la información - Los autoencoders

En esta sección, tras haber comprendido cómo se preprocesa la información procedente del sensor, se pretende entrar en la materia que supone la clave del trabajo, es decir, el empleo de sistemas de aprendizaje profundo para la detección de anomalías.

3.3.1. Estructura y parámetros de las redes

Esto es quizás lo más interesante desde un punto de vista del entendimiento de este sistema, pues la estructura elegida para la red neuronal es fundamental para conseguir el objetivo. Como se puede imaginar, esta estructura está basada en el concepto de autoencoder y hace uso de capas convolucionales para sacar un máximo partido a la correlación espacio-temporal dentro de las ventanas.

Además, la realidad es que, como ya hemos comentado, se hace uso de cuatro autoencoders distintos debido a que el comportamiento a lo largo de la fibra no es uniforme por lo que es bastante probable que utilizando estructuras diferentes de capas para cada uno de los autoencoders se habría podido optimizar el funcionamiento del sistema en su totalidad, pero por simplicidad y dado el buen funcionamiento obtenido de esta forma, se ha utilizado una misma estructura para los cuatro, la cual se describe a continuación.

Cuando se habla de redes convolucionales consideramos más acertado hablar en términos conjuntos de capas y esto es debido a que, aunque cuando se habla sobre capas convolucionales, muchas veces se da por hecho que el pooling está incluido en la misma,

mientras que esta estructura se compone realmente de dos capas. Por eso, nos referimos de ahora en adelante a un par de capas cuando hablamos de la concatenación de una capa de cada uno de estos dos tipos.

En cuanto al número de pares de capas, se han empleado seis. El porqué de este número es, al igual que sucede con otros aspectos del sistema, que un número menor de capas resulta en un sistema menos eficaz, pero uno con mayor número de éstas resulta más lento y no proporciona mejoras significativas, además de aumentar el riesgo de caer en el overfitting, que se puede interpretar como que la red memoriza los datos del entrenamiento en lugar de generalizar y sacar conclusiones [31]. Así pues, tres de estos pares de capas son para el codificador y la otra parte para el decodificador. Aunque realmente no tiene por qué ser así, se ha decidido que el autoencoder sea simétrico en cuanto a su estructura y que codificador y decodificador no sólo tengan un mismo número de capas sino que además la cantidad de neuronas de las capas también sea la misma. El motivo principal de esto es que es más sencillo mantener un control de las dimensiones y en casos como este, en el que las ventanas con las que se trabajan no son excesivamente grandes causando esto cierta inflexibilidad, supone una ventaja fundamental. Además de estos seis conjuntos de capas, se emplea una última capa convolucional que permite volver al tamaño de la imagen de entrada que es el fin último de un autoencoder. Todo esto es lo que se muestra en el esquema de la figura 3.5.

Además, algo que no se ha comentado hasta el momento es el tema de cómo se ha normalizado la información para introducirla a la red neuronal, ya que, como se ha explicado en la sección 2.2.6, es una acción indispensable para la convergencia de la red, además de tener aquí una importancia relevante por lo que a continuación se comenta. Bien, se ha utilizado un escalado MinMax para cada ventana, lo que comprime (o extiende) todos los valores de la ventana a valores entre 0 y 1. Esto es especialmente interesante aquí porque de esta forma se le está proporcionando la falsa información a la red de que la amplitud de la señal es igual en todas las distancias. Esto no es un problema sino que es más bien una ventaja, ya que todas las ventanas tienen unas características más similares unas con otras, lo que permite a la red generalizar mejor.

3.3.2. Entrenamiento

Con la estructura de la red definida, el siguiente paso es entrenarla. Para esto y basándonos en lo explicado en la sección 2.4, se han empleado señales que llamaremos silencio y que se refieren a medidas tomadas en instantes de tiempo en los que no se produce ningún tipo de evento sobre la fibra y, por lo tanto, la señal propagada dentro de la misma no debería de presentar ningún comportamiento anómalo. Esto último, como ya hemos comentado, no es del todo cierto debido a que la información presuntamente sin eventos con la que se ha trabajado realmente presenta un comportamiento ligeramente anómalo en las partes de la fibra correspondientes a las zanjas.

Concretamente, para entrenar las redes se han empleado únicamente cinco minutos

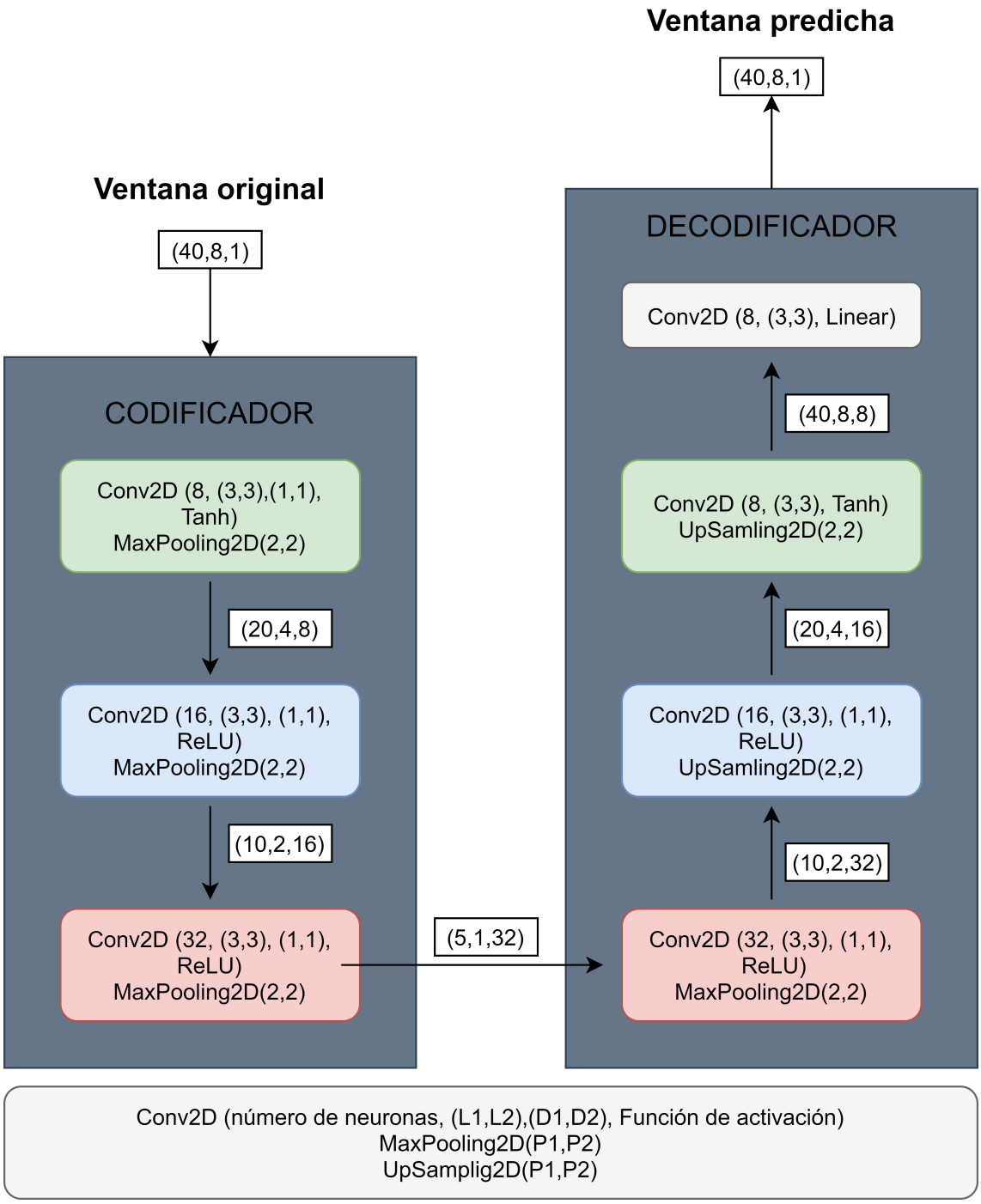


Fig. 3.5. Estructura de la red

de información de silencio, lo cual supone una ventaja fundamental del sistema, ya que en cuestión de pocos minutos de escucha, procesado y entrenamiento el sistema puede estar listo para llevar a cabo su cometido. Como es lógico, se hicieron pruebas basadas en entrenar las redes con más minutos de información de silencio pero no se observó una mejora clara en los resultados, por lo que se eligió cinco minutos como una cantidad de tiempo adecuada para el objetivo.

Posteriormente, en cuanto al optimizador, se eligió Adam con un learning rate de

0.001 debido a su bajo tiempo de convergencia, algo ideal para agilizar el entrenamiento. Además, proporciona unos resultados suficientemente buenos.

Después, la medida empleada como función de coste es el error cuadrático entre la entrada y la salida de la red por ser una medida que penaliza especialmente errores altos. Esto en la fase de predicción supondrá que una red entrenada con este criterio es sensible a desviaciones con respecto a los valores habituales, el cual es precisamente el objetivo del sistema.

Por último, en cuanto al tamaño de los batch y el número de epochs de las distintas redes, se muestran en la tabla 3.2 los valores empleados. En términos generales, un tamaño de batch menor permite a una red converger mejor, pero conlleva un tiempo de procesado superior. Todo es inversamente análogo en el número de epochs, es decir, cuanto mayor es este número, una red suele converger también mejor, pero penalizando el tiempo. Esto explica el porqué de los valores que aparecen en la tabla y es que, al ser las distancias más alejadas de la fibra las más difíciles de tratar, es conveniente disminuir el tamaño del batch y aumentar el número de epochs a costa de penalizar algo el tiempo de entrenamiento, mientras que para las distancias más cercanas podemos ser menos exigentes.

	Autoencoder 1	Autoencoder 2	Autoencoder 3	Autoencoder 4
Tamaño de batch	256	128	64	64
Número de epochs	20	30	40	50

TABLA 3.2. TAMAÑO DE BATCH Y NÚMERO DE EPOCHS DE CADA AUTOENCODER

3.3.3. Predicción

Tras estar los cuatro autoencoders entrenados, el sistema global está listo para funcionar y ser capaz de interpretar la posición e instantes de tiempo de posibles elementos anómalos. Esta parte en sí no presenta demasiado interés, ya que consiste únicamente en, tras preprocesar la información de los receptores, pasarla como entrada a la red entrenada y que ésta trate de reconstruirla. Lo que sí resulta de más interés es la sección siguiente del capítulo en la que se explica cómo se intenta sacar partido a la predicción realizada por la red.

3.4. Posprocesado de la información

En esta sección se va a terminar de explicar todo el procedimiento del sistema que pasa de unas medidas tomadas por el DAS a una matriz que contiene información sobre si en una posición espacio-temporal hay una anomalía. A continuación se explica cómo se pasa de la salida de los autoencoders a la matriz final, que es binaria por contener exclusivamente la información sobre si hay o no anomalías.

3.4.1. Reconstrucción de la matriz

La primera parte es la de reconstruir la matriz, ya que no hemos de perder de vista que en este momento la unidad de trabajo son las ventanas, pero nos interesa recuperar una señal con un formato idéntico al de la señal bidimensional que se obtiene tras llevar a cabo el primer paso del preprocesado y así compararla con la original. En concreto esto no tiene apenas interés conceptual y es un mero procedimiento que se ha de realizar de cara a ser capaces de tener dos señales que se puedan comparar.

3.4.2. Medida del error

Tras haber pasado del conjunto de ventanas que se obtienen a la salida del autoencoder a una matriz de formato idéntico a la entrada, la siguiente decisión que se ha de tomar es la de qué medida de error se va a emplear para comparar ambas matrices, el cual es el fin último del sistema. Concretamente, para este caso lo que se propone, debido a su simplicidad y al buen rendimiento que se obtiene, es el error cuadrático instantáneo. Además de esto, el hecho de haberse elegido el error cuadrático como función de coste en el entrenamiento supone que el error cuadrático instantáneo en las partes de la señal sin anomalías será especialmente bajo.

3.4.3. Decisor

Esta es la parte más interesante de esta sección, a pesar de pertenecer a otra rama de estudio diferente al tema principal de este trabajo, por dos motivos fundamentales. El primero es que se basa en criterios básicos pero fundamentales de la teoría de la decisión y el segundo es que se da por finalizado el sistema y se obtiene una salida binaria que indica si hay un evento o no lo hay. Debido a su relevancia se incluye un diagrama de bloques que ayuda a su comprensión en la figura 3.6.

Debido a no ser el tema principal del estudio, pero tener relevancia en el sistema final, en el anexo C se explican los conceptos básicos necesarios para comprender esta sección.

Primero, es importante aclarar que, aunque no se emplean sistemas basados en redes neuronales, lo aquí realizado sí comparte un cierto símil con éstos en cuanto a la metodología de trabajo, ya que, primero se produce una fase que se correspondería con la de entrenamiento, en la que se obtienen ciertos parámetros característicos de la señal para distintas distancias haciendo uso de las señales silencio para luego con estos parámetros tratar de estimar una función de densidad de probabilidad que modele de forma lo más acertada posible lo que llamaremos suelo de ruido en cada distancia. El nombre de suelo de ruido hace referencia al error residual que, a pesar de emplear señales silencio para la predicción, siempre prevalece y el motivo porque se le da este nombre es que se puede corresponder a la noción general de ruido en sistemas de comunicaciones y que supondremos inevitablemente presente en cualquier predicción. Después de esta fase que hemos

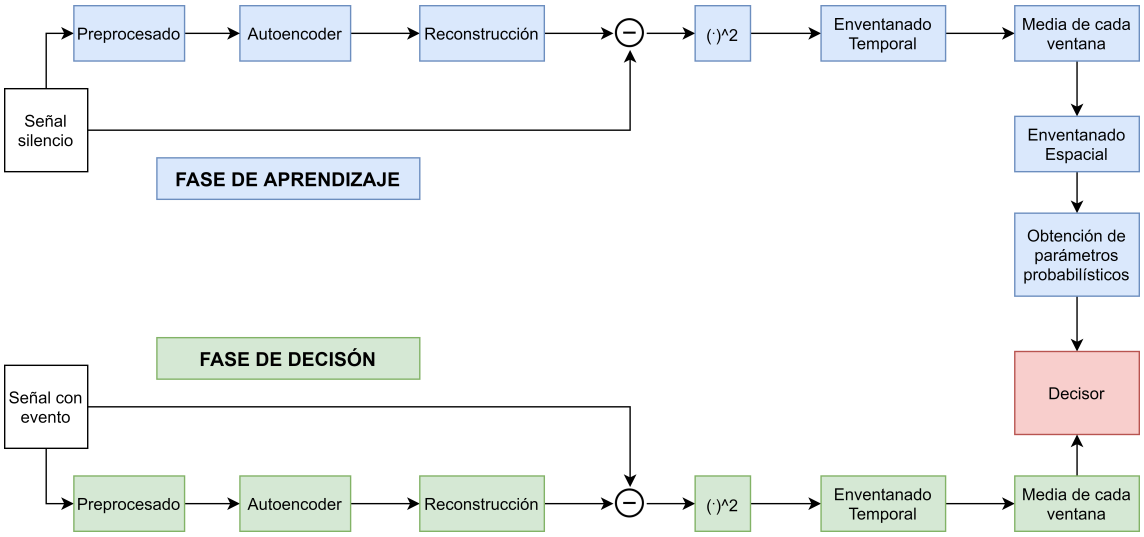


Fig. 3.6. Esquema del decisor

relacionado con la entrenamiento (es más apropiado llamarla de aprendizaje), el siguiente paso es finalmente lo que llamamos fase de decisión y que consiste lógicamente en decidir si hay un evento o no en base a las estadísticas obtenidas en la fase anterior. Estas dos fases convergen en un decisor que utiliza los resultados de las dos para devolver el veredicto final.

Las señales con las que vamos a trabajar en la fase de aprendizaje son señales de silencio, esto significa que vamos a pasar como entradas a los cuatro autoencoders los datos correspondientes a una captura en la que no hay eventos para obtener el error cuadrático medio entre entrada y reconstrucción en estos casos. Una vez obtenido, el objetivo es modelar en términos probabilísticos este error cuadrático medio.

Por lo tanto, partimos de una base en que la matriz del error para un minuto es de 6000×6000. Como se ha comentado ya, los últimos 2.8 km han de ser desechados, lo que supone quedarse con una matriz de 6000×5562 (recordamos que la resolución espacial es de 6.4 km) y por simplificar esta fase hemos decidido reducirlo a una de 6000×5550. Esta matriz, posteriormente se enventana en su dimensión temporal con unos valores de $N = 200$ y $M = 100$, lo cual supone, dada la resolución temporal, que cada una de las ventanas se desplaza un segundo. Posteriormente se calcula el valor medio de cada una de estas ventanas dando resultado a una matriz de 60x5550. Esto, evidentemente, supone que a partir de este momento la resolución temporal pasa a ser de un segundo. El porqué de esta decisión es simplemente que permite estimar mejor las estadísticas que a continuación explicaremos y, a su vez, una resolución mayor, en principio, y para las aplicaciones que se llevan a cabo en este momento no es necesaria. Resumiendo, esto significa que para la decisión se va a emplear como único parámetro el error cuadrático medio de cada ventana de un segundo entre la señal a entrada y salida de la red. Lo que hasta ahora hemos llamado B se correspondería por tanto con el hecho de que el error cuadrático medio de las ventanas de un segundo es igual a un valor concreto.

Una vez realizada esta transformación se realiza a su vez otro enventanado, pero en este caso en la dimensión espacial y con parámetros $N = 1110$ y $M = 555$. El motivo de esto es que, como ya hemos explicado en varias ocasiones a lo largo de este texto, el comportamiento de la fibra en reposo varía con la distancia a la que se encuentra del receptor acústico, por lo que la extracción de parámetros estadísticos del suelo de ruido se ha de realizar según partes de la fibra ya que varían significativamente a lo largo de ésta. El desplazamiento aquí realizado es de 555 muestras, lo que supone que los parámetros cambian cada 3.55 km. En este punto, por lo tanto, tenemos diez ventanas de 60×1110 cada una correspondiente al suelo de ruido en los segmentos que se describe en la tabla 3.3. La última ventana está duplicada, o visto de otra forma, hay nueve ventanas y la última se emplea para estimar el suelo de ruido de la distancia que se correspondería con las dos últimas ventanas. Esto es simplemente porque para estimar el suelo de ruido de cada segmento de 3.55 km se utilizan además de esos 3.55 km también los 3.55 km siguientes y, como es obvio, en el final de la fibra eso no se puede realizar, por lo que se emplean los 3.55 km anteriores.

Ventana	**Distancia en la señal ruido (km)**	**Distancia para las que se utiliza (km)**
1	0 - 7.10	0 - 3.55
2	3.55 - 10.65	3.55 - 7.10
3	7.10 - 14.20	7.10 - 10.65
4	10.65 - 17.75	10.65 - 14.20
5	14.20 - 21.3	14.20 - 17.75
6	17.75 - 24.85	17.75 - 21.30
7	21.30 - 28.40	21.30 - 24.85
8	24.85 - 31.95	24.85 - 28.40
9	28.40 - 35.50	28.40 - 31.95
10	28.40 - 35.50	31.95 - 35.50

TABLA 3.3. DISTANCIAS EMPLEADAS PARA ESTIMAR EL SUELO DE RUIDO

Una vez explicadas todas estas decisiones de diseño, queda ver cómo afrontar la estimación de parámetros estadísticos. Para ello, como se explica en el anexo C, lo que habitualmente se realizaría es estimar una función de densidad de probabilidad a posteriori para las dos hipótesis del problema, es decir, ausencia o presencia de eventos, llamándolas H_1 y H_2, respectivamente. Este es uno de los puntos en los que la falta de medidas entorpece el desarrollo ortodoxo y académicamente correcto del sistema. El porqué de esto es que para la realización de este trabajo apenas se ha contado con señales con presencia de anomalías, por lo que extraer unos parámetros de un conjunto de datos con tan reducido tamaño puede resultar incluso contraproducente. Según lo desarrollado en la expresión C.6 se debe de considerar que ha ocurrido una anomalía si y sólo si:

$$P(B|H_1) \leq P(B|H_2)\frac{\lambda_{12}}{\lambda_{21}}\frac{P(H_2)}{P(H_1)} \tag{3.1}$$

En la expresión 3.1 se ve perfectamente los problemas explicados en el párrafo anterior y es que no es posible estimar $P(B|H_2)$ por la falta de datos y además es complicado trabajar con unas probabilidades de los sucesos H_1 y H_2, ya que en otra situación en la que se contara con más información se podrían estimar dichos valores, pero en este caso de nuevo la cantidad de medidas es insuficiente. Por lo tanto hemos reducido el problema a algo que es común en estas situaciones y que consiste en obtener unos parámetros de probabilidad de B bajo la hipótesis H_1 y asumir que los datos que resultan improbables en esta función de densidad deben haberse producido por la presencia de una anomalía. Esto es sencillamente la expresión siguiente

$$P(B|H_1) \leq \gamma \tag{3.2}$$

Falta entonces explicar cómo obtener los valores de $P(B|H_1)$ y qué γ emplear. Pasemos primero con $P(B|H_1)$. Para ejemplificar esto, vamos a recurrir a la figura 3.7. Supondremos que la gráfica azul se corresponde con el error cuadrático medio en casos en los que no hay evento y la roja para los casos en los que sí que hay evento (aunque en realidad no la hemos calculado). Por lo tanto, la zona verdaderamente conflictiva a la hora de modelar estas funciones de probabilidad es la que se corresponde con los puntos cercanos al lugar en el que ambas se cruzan, suponiendo esto que la correspondiente al caso sin eventos no es necesario que sea modelada para valores bajos. Con fundamento en esta idea es como se ha trabajado aquí y es que lo que se ha realizado es descartar los datos del error cuadrático medio del suelo de ruido inferiores a un percentil y trabajar con el resto. Concretamente y tras probar distintos valores, se vio que el percentil 50 funcionaba suficientemente bien ya que, aunque podría parecer que cuanto mayor fuera el percentil mejor modelaría la parte de los valores altos de la función de densidad, como contrapartida aparece el hecho de que se cuenta con menos datos para estimar estas probabilidades. Por lo tanto, el resumen es que el objetivo es tratar de caracterizar la función de densidad de probabilidad que describe cómo se distribuye el error cuadrático medio para los casos superiores al percentil 50 cuando no hay eventos.

En este punto hay que explicar cómo se ha tratado de modelar la distribución del error cuadrático medio para los casos en los que no hay eventos. Para ello se muestran dos histogramas en la figura 3.8 correspondientes a una ventana de suelo de ruido. El de la 3.8a representa todos los datos y el de 3.8b los superiores al percentil 50 y que son con los que vamos a trabajar. Vemos fácilmente que una función de densidad de probabilidad que podría modelar bien los datos superiores al percentil 50 es la exponencial. Por lo tanto, se ha empleado el método fit de Scipy para la función expon [32], que realiza una estimación de máxima verosimilitud [33] y que devuelve los dos parámetros de la función de probabilidad exponencial que modela a los datos de cada una de las ventanas. En la figura 3.9 se representan superpuestos la función de densidad obtenida y el histograma de los datos para cuatro ventanas. En esta figura se entiende perfectamente el razonamiento que hemos seguido para calcular el suelo de ruido en ventanas en lugar de suponer que a lo largo de los 35.6 km de la fibra útil el ruido es invariante. Vemos claramente la diferencia

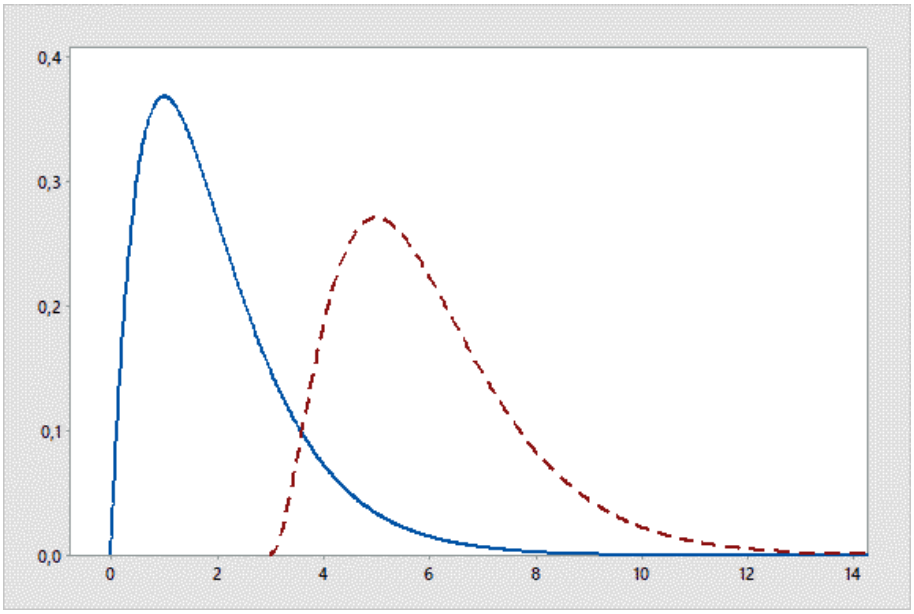
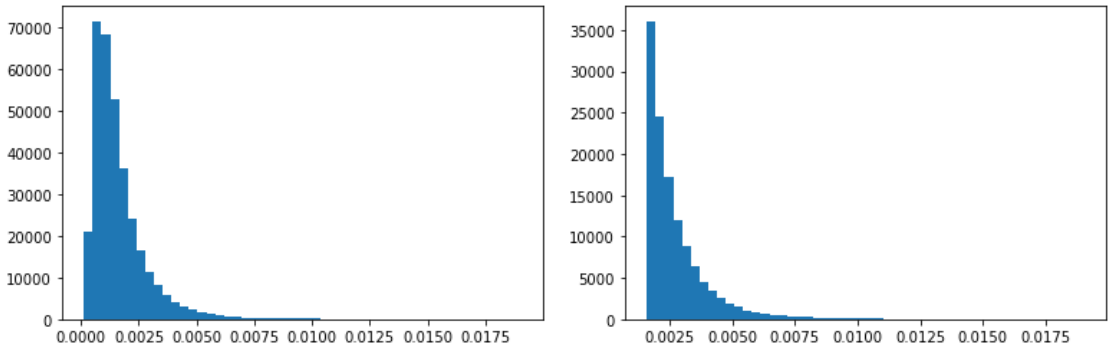


Fig. 3.7. Dos distribuciones de probabilidad

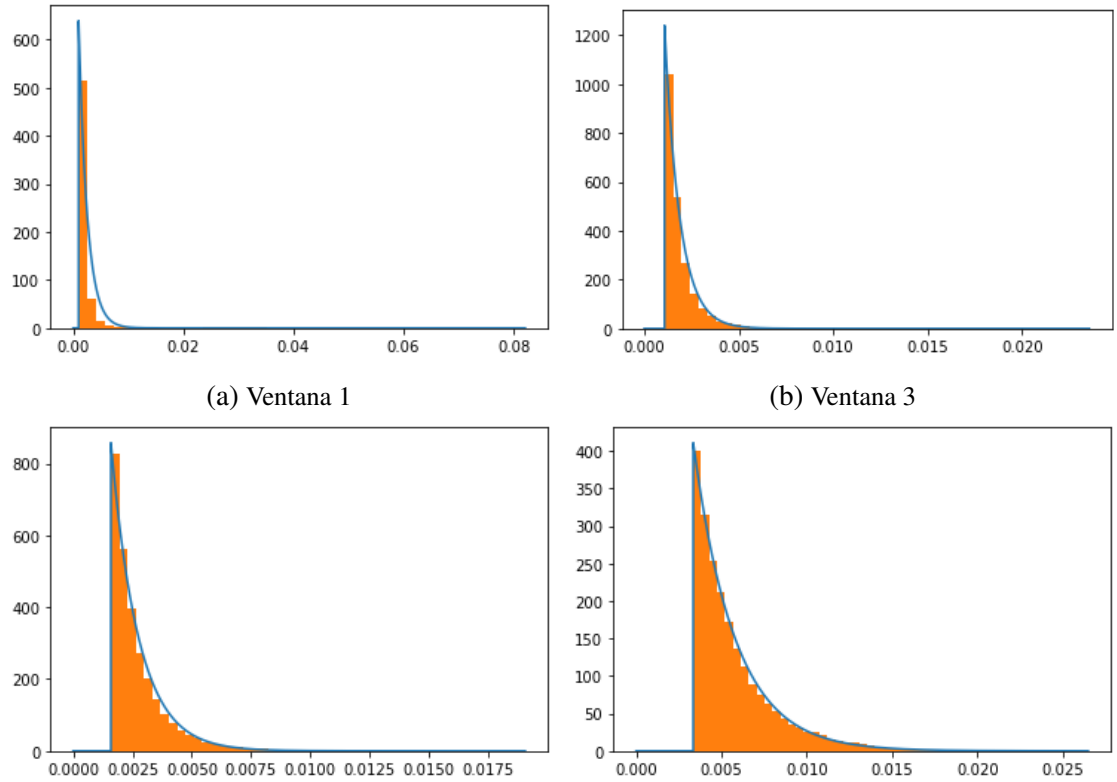
en las funciones de densidad de probabilidad en estas cuatro ventanas y que, como cabe esperar, el error de predicción tiene una potencia mayor cuanto mayor es la distancia a la que se encuentra del receptor acústico.



(a) Histograma de todos los datos (b) Histograma de datos superiores al percentil 50

Fig. 3.8. Comparación de las matrices entrada y salida esperada del sistema

Puede caber entonces la duda de qué valor de γ se ha de usar y es que es precisamente un parámetro de diseño, ya que en función de él dependen dos conceptos que aparecen explicados al final del anexo C y que describen el comportamiento final del sistema. Éstos son la probabilidad de falsa alarma y de detección. De hecho, qué valor de γ elegir es algo que ni si quiera vamos a mencionar aquí, ya que no es una decisión que debamos de tomar nosotros y además no aporta ninguna información del sistema y lo único que debe de quedar claro es que cuanto menor sea γ menor será la probabilidad de falsa alarma, pero también la de detección, y viceversa. Por lo tanto, la forma de presentar el rendimiento del sistema será mediante curvas ROC y se presentan en el siguiente capítulo. Esto se correspondería con el bloque de Decisor que aparece en la figura 3.6.



(a) Ventana 1

(b) Ventana 3

(c) Ventana 6

(d) Ventana 10

Fig. 3.9. Histograma y función de densidad del suelo de ruido en cuatro ventanas

En este punto falta explicar simplemente la fase de decisión, algo que en este punto no es un desafío, ya que consiste simplemente en realizar parte del proceso realizado en la fase de aprendizaje, pero aplicado a la señal que es susceptible de contener eventos. Concretamente se realiza hasta la parte correspondiente a calcular el error cuadrático medio de ventanas de un segundo ya que se va a tomar la decisión en función de cómo de verosímil es este ECM.

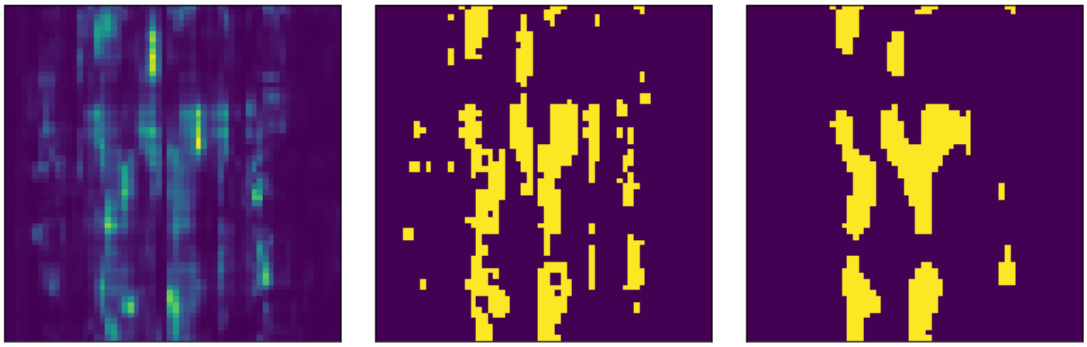
3.5. Posprocesado de la señal decisión

Tras haber explicado el decisor implementado, surgen algunos problemas que impiden medir de forma óptima las probabilidades de detección y falsa alarma. Solucionar estos problemas de forma completa es en sí una tarea complicada y lo aquí propuesto, igual que ocurre con el decisor, es simplemente una primera aproximación para tratar de dar un sistema lo más completo posible.

La raíz de este asunto se entiende muy bien si vemos de cerca el error de predicción en situaciones con evento, como en las figuras 3.10a, 3.11a y 3.12a. Aunque se podría pensar que el error de predicción de un evento es uniforme, aquí se ve claramente que no lo es y varía bastante. El motivo de esto realmente es desconocido, pero lo que sí que se sabe es que para trabajar con un decisor robusto habrá que procesar la salida del mismo.

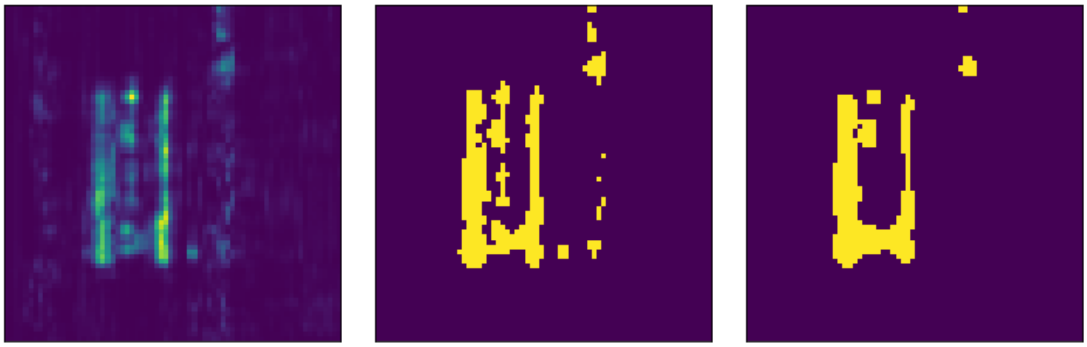
Además de que el error es poco uniforme, se ve también que en las zonas donde hay error, es decir, en las zanjas, hay una zona en medio donde el error es bajo, algo que se ha tenido en cuenta de cara a calcular las probabilidades de error.

Para tratar de paliar en cierta medida este efecto, se va a emplear un algoritmo sencillo para tratar la matriz salida del detector. Para ayudar a comprender esto, vamos a elegir los tres eventos de las figuras 3.10, 3.11 y 3.12, que son de naturalezas distintas y vamos a mostrar la señal entrada al decisor, la salida del mismo (que es la entrada del algoritmo) y la salida del algoritmo. Cabe recalcar que, aunque aquí se ha tomado un segmento de la señal como entrada del algoritmo para ver más claramente su funcionamiento, está pensado para trabajar con las señales completas de la salida del decisor.



(a) Entrada del decisor (b) Salida del decisor (c) Salida del algoritmo

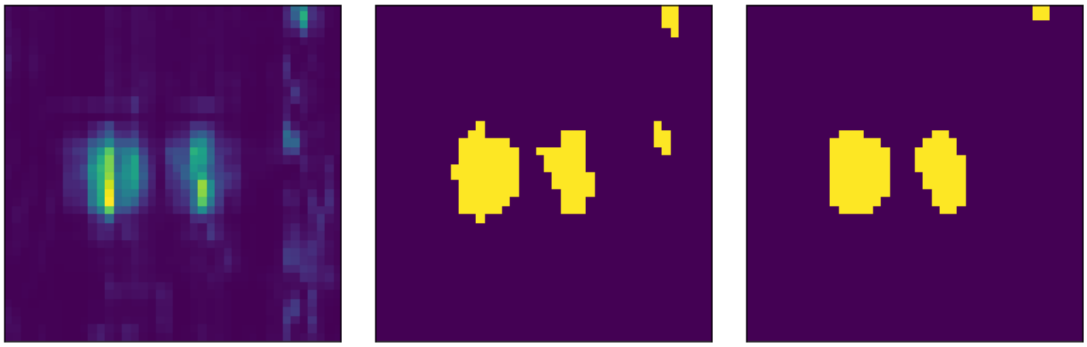
Fig. 3.10. Señales correspondientes a un evento producido por excavadora avanzando



(a) Entrada del decisor (b) Salida del decisor (c) Salida del algoritmo

Fig. 3.11. Señales correspondientes a un evento producido por oruga

En estas figuras se ven varias cosas. La primera, el funcionamiento del decisor propuesto en la sección anterior, que vemos que funciona correctamente, pero da una señal ruidosa por el propio comportamiento de la señal de entrada. El algoritmo de posprocesado de la señal decisión lo que realiza es tratar de limpiar en cierta medida la señal de salida del decisor para que sea más uniforme para que posteriormente, interpretar la ocurrencia de un evento sea más sencillo.



(a) Entrada del decisor (b) Salida del decisor (c) Salida del algoritmo

Fig. 3.12. Señales correspondientes a un evento producido por martillo hidráulico

En cuanto al funcionamiento del algoritmo es bastante simple. Dada la entrada $X = (x_{ij}) \in \mathbb{R}^{M \times N}$ la salida $Y = (y_{ij}) \in \mathbb{R}^{M \times N}$ está dada por la ecuación 3.3.

$$y_{ij} = Round\left(\frac{1}{65}\sum_{m=i-2}^{i+2}\sum_{n=j-2}^{j+2} a_{mn}x_{mn}\right) \text{ para } 2 < i < M-1 \text{ y } 2 < j < N-1$$
$$y_{ij} = x_{ij} \text{ en caso contrario} \tag{3.3}$$

Donde los coeficientes a_{mn} vienen dados en la figura 3.13.

	i-2	i-1	i	i+1	i+2
j-2	1	2	3	2	1
j-1	2	3	4	3	2
j	3	4	5	4	3
j+1	2	3	4	3	2
j+2	1	2	3	2	1

Fig. 3.13. Coeficientes del algoritmo de procesado de la salida del decisor

El algoritmo consiste simplemente en obtener la posición $i\,j$ de la salida como la media ponderada de los valores que rodean a esa posición. Cuanto más lejos está un valor de la posición $i\,j$ menor influencia tiene sobre él.

4. RESULTADOS EXPERIMENTALES

En este capítulo se van a presentar los resultados experimentales y para ello vamos a dividir en dos secciones el capítulo. En la primera veremos los mapas tiempo-distancia del error de predicción, que son los que mejor describen la mayor parte del trabajo realizado. Por otro lado, dado que hemos diseñado un sistema de decisión, en la segunda parte del capítulo daremos unas curvas ROC, ya que son una forma sencilla de analizar los resultados obtenidos.

4.1. Mapas tiempo-distancia del error de predicción

Con todo esto, vamos a comenzar mostrando en forma de mapa tiempo-distancia algunas de las medidas de error para distintos eventos a continuación y explicar un poco el porqué de algunos aspectos. Todos ellos son correspondientes a medidas cuyas fichas aparecen en el Anexo D, por lo que se puede corroborar que, efectivamente, las zonas con un error mayor se corresponden con los instantes y posiciones que deben.

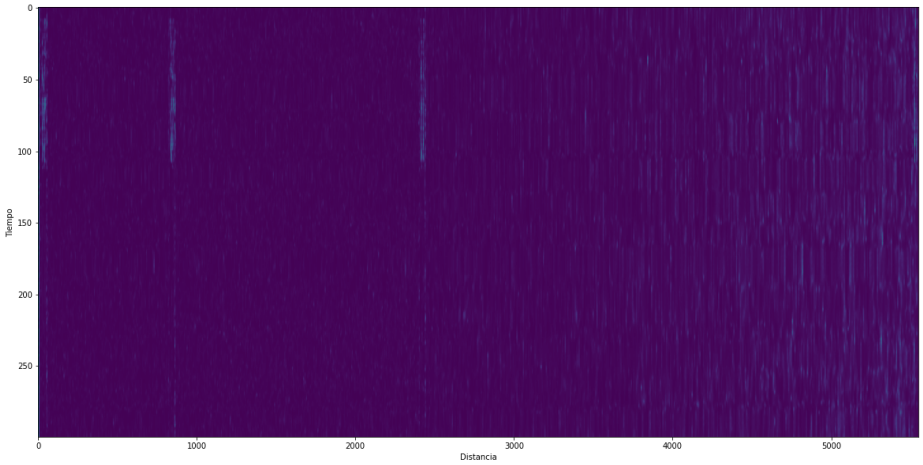


Fig. 4.1. Entra: Error de predicción de excavadora avanzando sobre la fibra

En estos mapas vemos dos cosas muy claras. La primera es que, como se ha venido comentando y como cabría esperar antes de adentrarnos en el estudio de este sistema, la señal del error para las distancias más alejadas del receptor acústico son significativamente más ruidosas. Esto se traduce en que eventos como un martillo hidráulico picando sobre el terreno en el que está enterrado la fibra óptica (4.5) son detectables sin problema, pero en cambio, eventos como una excavadora cavando y añadiendo tierra (4.4) no lo son. Por otro lado, en distancias relativamente cercanas (menos de 20 km) todos los eventos con los que

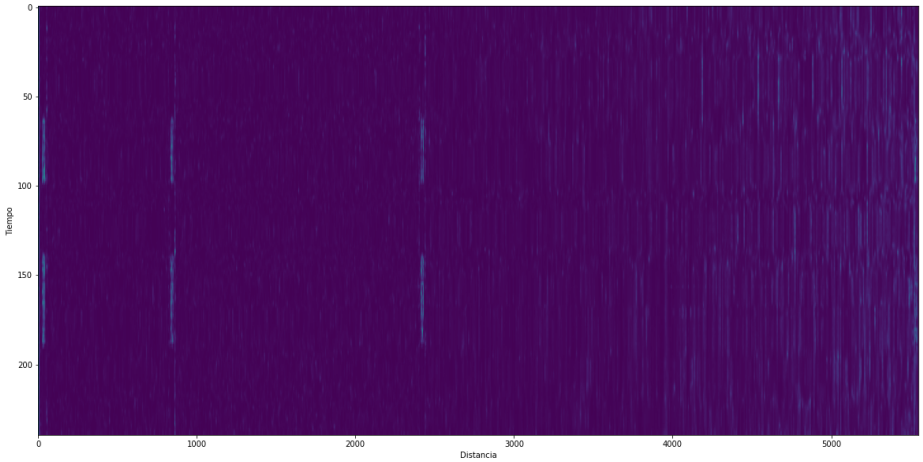


Fig. 4.2. Oruga 0m: Error de predicción de excavadora picando sobre la fibra

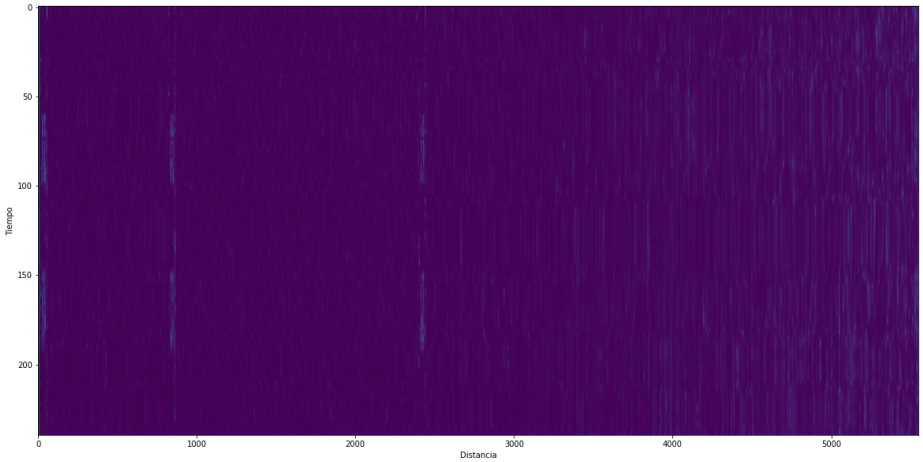


Fig. 4.3. Oruga 10m: Error de predicción de excavadora picando a 10 metros de la fibra

se ha trabajado son fácilmente detectables. La segunda es que si comparamos las figuras 4.2 y 4.3 podemos ver que cuando una acción no se produce exactamente sobre el terreno de la fibra, sino que se produce a una distancia, como diez metros, aunque la magnitud del error disminuye sigue siendo detectable. Además, una excavadora avanzando sobre la fibra causa un impacto irregular (4.1), algo que puede ser debido, por ejemplo, al cambio de velocidad.

Vamos a aprovechar aquí para comentar algo que ya hemos mencionado en la sección 3.1 y es que debido a el escenario de captura de datos, las zonas de zanja, aún en casos en los que no hay eventos tienen un error de predicción mayor, puesto que el comportamiento de la luz de la fibra no es el mismo en los carretes que en la zanja. En la figura 4.6 se muestra un mapa tiempo-distancia para una medida de 5 minutos en la que no se

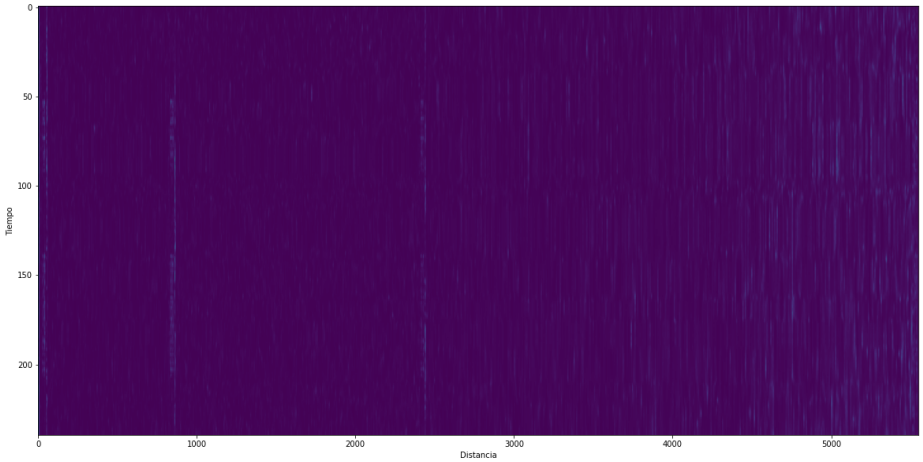


Fig. 4.4. Cazo 0m: Error de predicción de excavadora cavando y tapando sobre la fibra

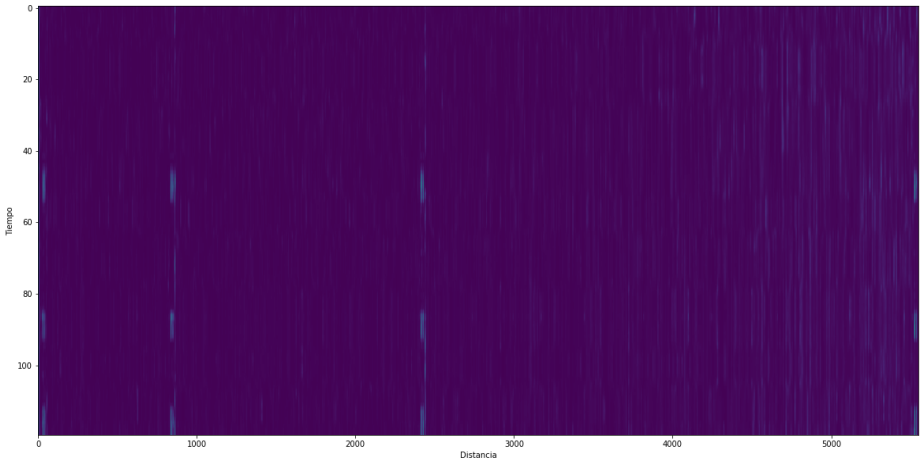


Fig. 4.5. Martillo hidráulico iteración 1: Error de predicción de martillo hidráulico picando sobre la fibra

produce ningún evento. Como se puede ver, las partes correspondientes a la zanja tienen un error cuadrático medio superior. En este punto se podría haber considerado dedicar autoencoders a entrenarse exclusivamente para las zanjas, pero hemos contado para realizar este trabajo con sólo 15 minutos de medidas de silencio, lo que supone una cantidad insuficiente para plantearse entrenar autoencoders para zonas tan específicas. Realmente, la magnitud de este error no es demasiado elevada comparado con las inmediaciones pero es algo que merece la pena comentar. Esto, en un escenario en el que toda la fibra estuviera desplegada en un escenario homogéneo no ocurriría.

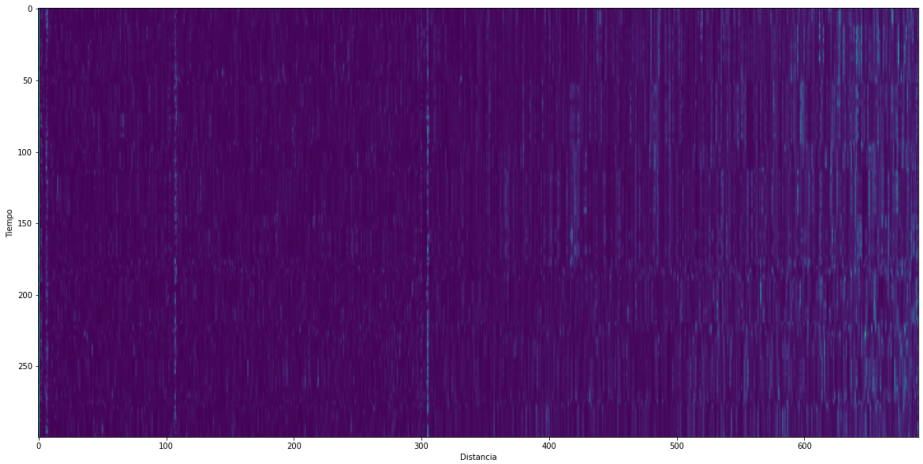


Fig. 4.6. Error de predicción para una medida tomada en situación de ausencia de eventos

4.2. Curvas ROC para algunas señales

En esta sección se presentan las curvas ROC de algunas señales con eventos. Este procedimiento lo hemos llevado a cabo exclusivamente para señales con eventos que producen un error de predicción más o menos uniforme porque a pesar de que el algoritmo explicado en la sección 3.5 funciona bien tratando de limpiar de residuos la señal que sale del decisor. A pesar de esto, por ejemplo, para el caso de la figura 3.10 sigue resultando complicado calcular las probabilidades de falsa alarma y de detección, ya que aunque estamos interpretando el evento de una excavadora avanzando como uno único, la realidad es que la señal salida del algoritmo último presenta zonas de supuesto silencio donde en teoría hay un evento. Una posibilidad para solucionar este problema sería realizar un diezmado espacial, pero a cambio perderíamos resolución, algo que no nos ha interesado.

Como comentarios particulares de estas curvas, la 4.7 es la que peores prestaciones presenta, algo que es normal debido a que una excavadora avanzando es un evento algo más irregular, tal y como se puede ver en la figura 4.1.

En cuanto a las de las figuras 4.8 y 4.9, las prestaciones son mejores que para el caso anterior y además, como cabe esperar, se obtiene un rendimiento ligeramente mejor para el caso en que se actúa directamente sobre la fibra.

Por último, la curva del martillo hidráulico de 4.10 es la mejor de todas, ya que como se ha visto en la figura 4.5 es el evento mecánico que mayor impacto produce en la forma de propagación de la luz dentro de la fibra.

Como comentario general, la relación entre probabilidades de detección y falsa alarma son bastante buenas, pero realmente no es la forma óptima de medir el trabajo realizado, ya que el sistema final de decisión es algo que tiene bastante trabajo por delante y que podría mejorar estas curvas significativamente manteniendo la misma estructura de sistemas

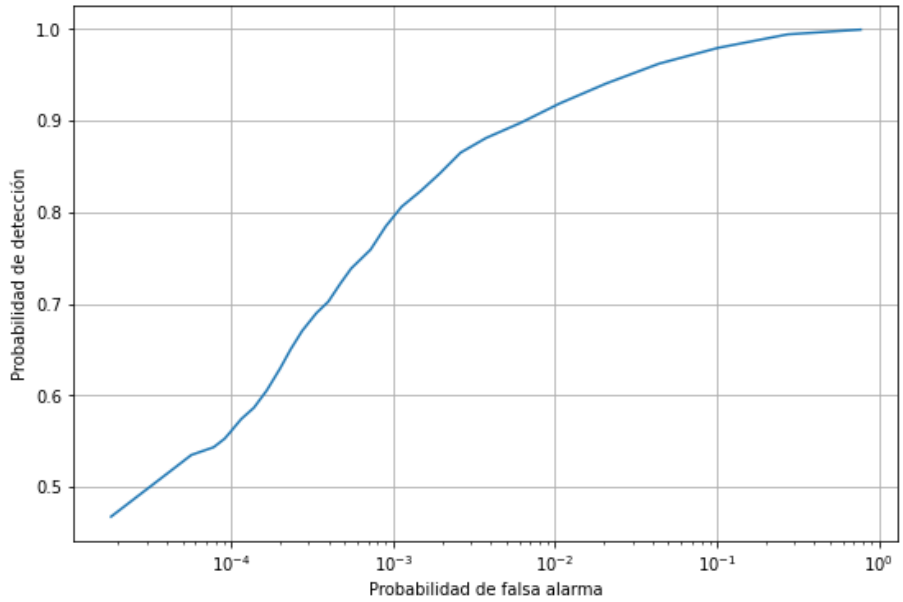


Fig. 4.7. Entra: Curva ROC de excavadora avanzando sobre la fibra

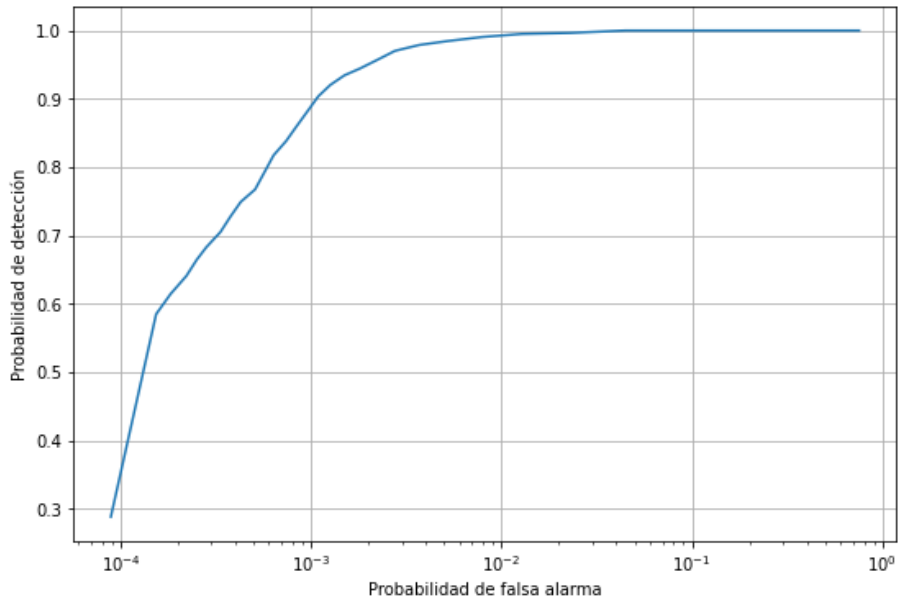


Fig. 4.8. Oruga 0m: Curva ROC de excavadora picando sobre la fibra

de aprendizaje profundo, el cual es el verdadero tema del trabajo.

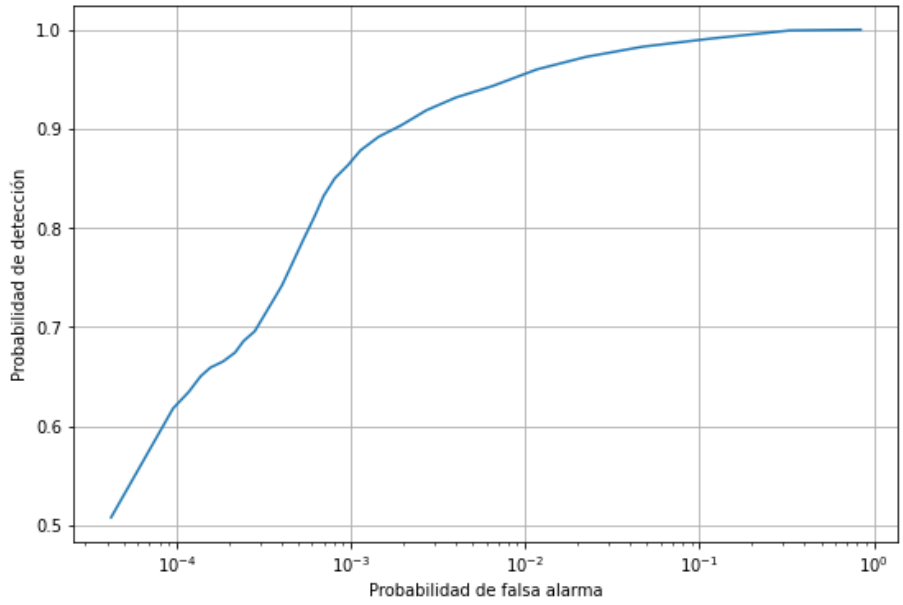


Fig. 4.9. Oruga 0m: Curva ROC de excavadora picando a 10 metros de la fibra

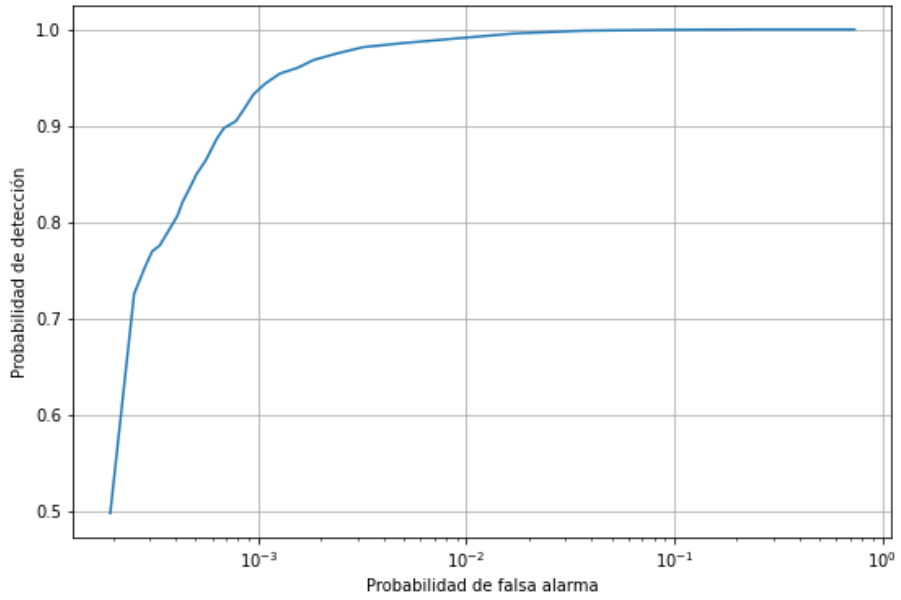


Fig. 4.10. Martillo hidráulico iteración 1: Curva ROC de martillo hidráulico picando sobre la fibra

5. CONCLUSIONES Y LÍNEAS FUTURAS

En este capítulo se pretende recoger las conclusiones que se pueden extraer de este trabajo, así como analizar el cumplimiento de los objetivos planteados inicialmente. También se comentarán algunas de las ideas que no se han llegado a implementar por una falta de tiempo y de un banco de medidas de mayor tamaño, pero que se espera puedan mejorar en cierta medida el rendimiento del sistema.

5.1. Conclusiones

Llegados a este punto y con todo el trabajo llevado a cabo y explicado, es indispensable tratar de resumir las conclusiones que se pueden extraer de este trabajo, así como de si se han cumplido los objetivos planteados.

La principal conclusión que se puede extraer de este trabajo es que se ha demostrado la viabilidad de un sistema para detectar eventos en señales procedentes de un DAS desde un enfoque de detección de anomalías utilizando técnicas de aprendizaje profundo, algo que supone una innovación, ya que es el primer acercamiento para resolver el problema de esta forma.

Por otro lado, se ha mostrado un sistema completo y funcional que, partiendo de una señal capturada por un DAS, es capaz de detectar la presencia de un evento con unas probabilidades de detección y falsa alarma bastante aceptables. Tanto el desarrollo del sistema como la caracterización del rendimiento del mismo son susceptibles de mejorar notablemente con una cantidad superior de medidas.

Uno de los objetivos iniciales era conocer algunas de las tecnologías de aprendizaje profundo, cosa que podemos decir que hemos cumplido. Esto no es algo objetivamente mensurable, pero hemos sido capaces de desarrollar un sistema basado en estas tecnologías y que cumple con los objetivos inicialmente planteados, por lo que podemos considerar que en este aspecto los resultados son satisfactorios. Por otro lado, en cuanto a la valoración del sistema, podemos ver varias cosas que en los siguientes párrafos comentamos.

Para empezar, lo más importante es que el sistema desarrollado es completamente funcional y dadas unas señales procedentes del DAS es capaz de decidir si hay o no un evento en una posición e instante concreto con unas probabilidades de detección y falsa alarma aceptables, a pesar de no ser ésta la mejor forma de caracterizar el rendimiento del sistema en general.

Otro aspecto fundamental de esto y que se consigue en parte gracias a haber empleado de técnicas aprendizaje profundo es el hecho de que el sistema descrito sería funcional en

otros entornos. En este aspecto, se puede decir que el sistema es bastante flexible y la idea de utilizar varios autoencoders para resolver el problema permite tratar de forma especial algunas situaciones potencialmente problemáticas.

Por otro lado, en cuanto al sistema decisor basado en la teoría de decisión bayesiana, su funcionamiento es aceptable y, a pesar de no ser el tema principal del trabajo, se ha resuelto el problema de forma sencilla y sin una dificultad conceptual demasiado alta resultando además computacionalmente eficiente, algo fundamental de cara a un sistema que pretende ser utilizado en tiempo real.

Estos párrafos se resumen en que los objetivos establecidos hace cuatro meses se han cumplido considerando además que en algunos aspectos los resultados han sido mejores de lo esperado y que el escenario que se ha desarrollado se corresponde con uno de los más optimistas que se plantearon al comienzo del trabajo.

5.2. Líneas futuras

El objetivo de esta sección no es tratar de dar ideas de aplicaciones del sistema descrito, ya que es algo que no es propio de este trabajo, sino que más bien es listar algunas ideas que por falta de tiempo o por estar completamente fuera del objetivo inicial del trabajo se han quedado en el tintero y no se han llegado a desarrollar.

- **Distintos autoencoders para distintas distancias.** Tal y como se ha explicado, la estructura de los cuatro autoencoders es exactamente igual, algo que ha sido así por simplicidad del trabajo y porque, aunque se hicieron algunas pruebas muy superficiales no se terminó de ver una clara mejora del rendimiento empleando estructuras distintas según partes de la fibra. Pero estas pruebas, desde luego, podrían haber sido más extensas y es posible que para las distancias más alejadas del receptor el uso de autoencoders con más capas resultara en una mejora en cuanto a la distinción de eventos.

- **Clasificación de eventos.** Aquí, debido principalmente a la falta de una gran cantidades de señales disponibles, se ha enfocado el trabajo desde un punto de vista más conceptual y que permite detectar anomalías. Sin embargo, para entornos en los que claramente haya un conjunto fijo de posibles tipos de eventos, basándonos y extendiendo lo explicado sobre detección de anomalías y teoría de decisión, y utilizando métricas quizás más apropiadas para esta tarea que el error cuadrático medio, se podría realizar un sistema clasificador de las anomalías detectadas.

- **Redes neuronales recurrentes.** En el campo de las redes neuronales hay un tipo de capas que permiten aprovechar información de entradas pasadas para trabajar con la entrada presente. Esto permite aprovechar una relación temporal que puede existir en series temporales. De aquí surgió una idea que consiste en agrupar ventanas como las descritas de forma que se pudieran ver como vídeos donde los fotogramas

son las ventanas, por lo que el problema se traduciría a encontrar anomalías en un vídeo [34, 35, 36]. La adición entonces de algunas capas recurrentes permitiría quizá mejorar el rendimiento de este sistema. De hecho, se hizo una prueba basada en esta misma idea y que consistía en utilizar redes convolucionales 3D, que permite trabajar de forma conjunta con ventanas consecutivas en el tiempo. Esto, aunque el sistema seguía siendo funcional, no mejoraba el rendimiento y además suponía un aumento del coste computacional. Por lo tanto, sería una solución más elegante y posiblemente daría un mejor rendimiento la utilización de capas recurrentes.

BIBLIOGRAFÍA

[1] J. Park y H. F. Taylor, "Fiber Optic Intrusion Sensor using Coherent Optical Time Domain Reflectometer," *Japanese Journal of Applied Physics*, vol. 42, n.º Part 1, No. 6A, pp. 3481-3482, 2003. DOI: 10.1143/jjap.42.3481. [En línea]. Disponible en: https://doi.org/10.1143%2Fjjap.42.3481.

[2] "¿Qué pasa en las vías del tren? Pregúntale a la fibra óptica," *Heraldo de Aragón*, 2017. [En línea]. Disponible en: https://www.heraldo.es/noticias/sociedad/2017/10/30/que-pasa-las-vias-del-tren-preguntale-fibra-optica-1204293-310.html.

[3] A. Owen, G. Duckworth y J. Worsley, "OptaSense: Fibre optic distributed acoustic sensing for border monitoring," en *2012 European Intelligence and Security Informatics Conference*, IEEE, 2012, pp. 362-364.

[4] T. M. Daley et al., "Field testing of fiber-optic distributed acoustic sensing (DAS) for subsurface seismic monitoring," *The Leading Edge*, vol. 32, n.º 6, pp. 699-706, 2013.

[5] T. Parker, S. Shatalin y M. Farhadiroushan, "Distributed Acoustic Sensing–a new tool for seismic applications," *first break*, vol. 32, n.º 2, pp. 61-69, 2014.

[6] Wikipedia, *Fibra oscura — Wikipedia, La enciclopedia libre*, [Internet; descargado 24-junio-2020], 2019. [En línea]. Disponible en: https://es.wikipedia.org/w/index.php?title=Fibra_oscura&oldid=117345730.

[7] A. T. Young, "Rayleigh scattering," *Applied optics*, vol. 20, n.º 4, pp. 533-535, 1981.

[8] J. Mateo, M. n. Losada e I. Garcés, "Limitaciones de las fibras ópticas," en *Dispositivos y Sistemas de Transmisión Óptica*, 2017, cap. 2, pp. 40-70.

[9] T. Horiguchi y M. Tokuda, "Optical time domain reflectometer for single-mode fibers," *IEICE TRANSACTIONS (1976-1990)*, vol. 67, n.º 9, pp. 509-515, 1984.

[10] Sanjay, *Difference between OTDR and COTDR*. 2016. [En línea]. Disponible en: https://mapyourtech.com/entries/general/difference-between-otdr-and-cotdr-.

[11] J. P. Garbayo, D. Sanahuja, C. Heras, J. Subías y Í. Salinas, "Desarrollo y caracterización de un sensor acústico distribuido basado en la técnica de medida C-OTDR," *Jornada de Jóvenes Investigadores del I3A*, vol. 6, 2018.

[12] W. S. McCulloch y W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, n.º 4, pp. 115-133, 1943.

[13] D. Nguyen y B. Widrow, "Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights," en *1990 IJCNN International Joint Conference on Neural Networks*, IEEE, 1990, pp. 21-26.

[14] J. Y. Yam y T. W. Chow, "A weight initialization method for improving training speed in feedforward neural network," *Neurocomputing*, vol. 30, n.º 1-4, pp. 219-232, 2000.

[15] C. E. Shannon, "A mathematical theory of communication," *Bell system technical journal*, vol. 27, n.º 3, pp. 379-423, 1948.

[16] A. Cauchy, "Méthode générale pour la résolution des systemes d'équations simultanées," *Comp. Rend. Sci. Paris*, vol. 25, n.º 1847, pp. 536-538, 1847.

[17] D. P. Kingma y J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[18] M. Shanker, M. Y. Hu y M. S. Hung, "Effect of data standardization on neural network training," *Omega*, vol. 24, n.º 4, pp. 385-397, 1996.

[19] G. K. Wallace, "The JPEG still picture compression standard," *IEEE transactions on consumer electronics*, vol. 38, n.º 1, pp. xviii-xxxiv, 1992.

[20] L. Theis, W. Shi, A. Cunningham y F. Huszár, "Lossy image compression with compressive autoencoders," *arXiv preprint arXiv:1703.00395*, 2017.

[21] V. Chandola, A. Banerjee y V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, n.º 3, pp. 1-58, 2009.

[22] N. Görnitz, M. Kloft, K. Rieck y U. Brefeld, "Toward supervised anomaly detection," *Journal of Artificial Intelligence Research*, vol. 46, pp. 235-262, 2013.

[23] J. MacQueen et al., "Some methods for classification and analysis of multivariate observations," en *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Oakland, CA, USA, vol. 1, 1967, pp. 281-297.

[24] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern recognition letters*, vol. 31, n.º 8, pp. 651-666, 2010.

[25] D. Pelleg, A. W. Moore et al., "X-means: Extending k-means with efficient estimation of the number of clusters.," en *Icml*, vol. 1, 2000, pp. 727-734.

[26] M. Sakurada y T. Yairi, "Anomaly detection using autoencoders with nonlinear dimensionality reduction," en *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, 2014, pp. 4-11.

[27] M. Schreyer, T. Sattarov, D. Borth, A. Dengel y B. Reimer, "Detection of anomalies in large scale accounting data using deep autoencoder networks," *arXiv preprint arXiv:1709.05254*, 2017.

[28] Y. LeCun y C. Cortes, "MNIST handwritten digit database," 2010. [En línea]. Disponible en: http://yann.lecun.com/exdb/mnist/.

[29] Wikipedia contributors, *Window function — Wikipedia, The Free Encyclopedia*, [Online; accessed 19-June-2020], 2020. [En línea]. Disponible en: https://en.wikipedia.org/w/index.php?title=Window_function&oldid=951701883.

[30] R. Ladelsky, "Matrix flattening and transposing in GCC," en *Proceedings of the GCC Developers' Summit*, 2006, pp. 97-98.

[31] D. M. Hawkins, "The problem of overfitting," *Journal of chemical information and computer sciences*, vol. 44, n.º 1, pp. 1-12, 2004.

[32] Scipy.org, *Statistical functions (scipy.stats.expon)*, 2019. [En línea]. Disponible en: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.expon.html.

[33] H. White, "Maximum likelihood estimation of misspecified models," *Econometrica: Journal of the Econometric Society*, pp. 1-25, 1982.

[34] T. Xiang y S. Gong, "Video behavior profiling for anomaly detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, n.º 5, pp. 893-908, 2008.

[35] V. Saligrama y Z. Chen, "Video anomaly detection based on local statistical aggregates," en *2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2012, pp. 2112-2119.

[36] Y. Zhao et al., "Spatio-temporal autoencoder for video anomaly detection," en *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 1933-1941.

[37] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, n.º 02, pp. 107-116, 1998.

[38] A. Krizhevsky, I. Sutskever y G. E. Hinton, "Imagenet classification with deep convolutional neural networks," en *Advances in neural information processing systems*, 2012, pp. 1097-1105.

A. FUNCIONES DE ACTIVACIÓN UTILIZADAS

En este anexo se recogen las funciones de activación utilizadas en este trabajo y que son también de las más utilizadas en la actualidad por razones que a continuación se comentan.

Para empezar, la función Linear es la más sencilla de todas y es que consiste simplemente en dejar la salida igual que la entrada.

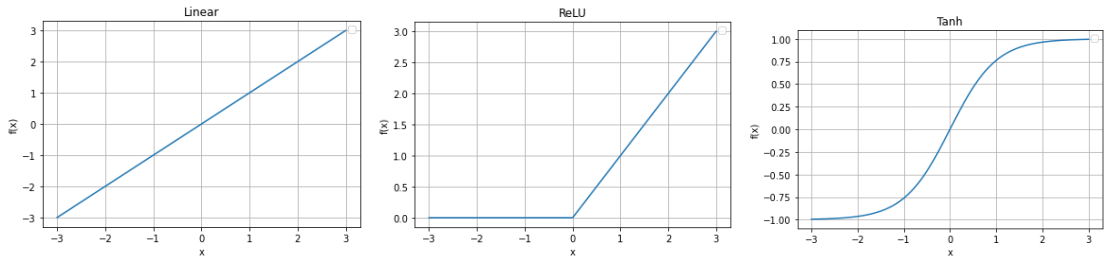
La función ReLU (Rectified Linear Activation Function) se define en la expresión A.1 y que está ganando mucha popularidad en los últimos años con el uso de redes con más capas. Como ventajas principales tiene que es computacionalmente eficiente y que además no sufre un problema conocido como desvanecimiento de gradiente, que consiste básicamente en que el valor del gradiente de la salida de la red va reduciéndose cuantas más capas se introducen [37].

$$ReLU(x) = max(0, x) \tag{A.1}$$

La función tanh (tangente hiperbólica) se define en la expresión A.2. Su salida está en el rango $[-1, 1]$ y es utilizada cuando se quiere mantener la salida de una capa en ese rango. Es común emplearla en la última capa de la red por este motivo.

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{A.2}$$

En la figura A.1 se muestra el comportamiento de forma gráfica de estas funciones de activación.



(a) Linear (b) ReLU (c) Tanh

Fig. A.1. Gráficas de las funciones de activación explicadas

B. CAPAS CONVOLUCIONALES DESDE EL PROCESADO DE SEÑALES

Como ya se ha explicado en la sección 2.2, el objetivo de las redes neuronales es emular en cierta medida los mecanismos que los humanos tienen para adquirir nuevos conocimientos y uno de estos es la vista. Es por esto que se emplean redes neuronales que aprovechan las características de las imágenes o, alternativamente, señales de dos dimensiones, que es precisamente el caso de este proyecto [38]. Esto resulta en que lo que acabamos de explicar en la sección anterior no es la única forma de conectar las neuronas en una red neuronal, sino que es simplemente una de las más sencillas e intuitivas por lo que su explicación permite comprender el concepto general detrás de estos sistemas de aprendizaje profundo, pero realmente en este trabajo la forma de conectar las neuronas no va a ser la descrita. El nombre que se les da a estas redes neuronales es convolucionales (CNN), por razones que se van a explicar a continuación.

En matemáticas, una convolución es un operador que transforma dos funciones x y h en otra función y, que es una superposición de x y una versión invertida y trasladada de h, o viceversa, ya que es un operador conmutativo. Cuando estas funciones son $\mathbb{Z} \rightarrow \mathbb{R}$ nos referimos a ellas como señales discretas o vectores, tal y como hemos venido haciendo hasta ahora. En este caso la convolución se define según la expresión B.1.

$$y[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k] \tag{B.1}$$

Este es un operador ampliamente utilizado en las telecomunicaciones entre otras cosas porque caracteriza el cambio que sufre una señal x tras atravesar un sistema lineal e invariante en el tiempo (LTI) h. Los sistemas LTI son fundamentales debido a que caracterizan gran cantidad de procesos físicos y además pueden ser analizados en detalle y ofrecer gran cantidad de información y herramientas poderosas que conforman el núcleo del procesado de señales. Las funciones que intervienen en una convolución suelen recibir el nombre de señal de entrada (x), filtro o kernel (h) y señal de salida (y).

La convolución puede extenderse a funciones $\mathbb{Z}^2 \rightarrow \mathbb{R}$, lo que denominamos señales bidimensionales, imágenes o matrices, dando lugar a la expresión B.2. Esto es perfectamente extensible a señales n-dimensionales, pero por la forma de abordar el problema no es necesario recurrir a ellas.

$$Y[m,n] = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} H[k_1,k_2]X[m-k_1,n-k_2] \tag{B.2}$$

Aunque esta definición de convolución es también válida para señales de duración finita, es una expresión general que puede ser confusa e innecesaria cuando se está trabajando con señales finitas, como es este caso. Es por eso que se emplea la expresión B.3

para casos en los que X o H tienen una longitud limitada. Aunque en este trabajo ambas señales son de duración finita por el simple hecho de estar trabajando con un ordenador, las que van a limitar en este aspecto serán los filtros H, puesto que el tamaño de estos es menor que el de las imágenes de entrada X. Es por eso que a L_1 y L_2 nos referiremos como longitudes del filtro horizontal y vertical, respectivamente. Además cada uno de los $H[k_1, k_2]$ son llamados coeficientes del filtro quedando, por tanto:

$$Y[m,n] = \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} H[k_1,k_2]X[m-k_1+1, n-k_2+1] \tag{B.3}$$

Esta definición clásica de convolución no es de la que realmente hacen uso las capas convolucionales. Es, sin embargo, otro operador matemático similar, pero con distinta definición el que realmente emplean estas las CNN, el cual se describe en la expresión B.4 para una entrada $X \in \mathbb{R}^{M_x \times N_x}$. A partir de ahora nos referiremos a él como convolución aunque realmente su nombre es correlación cruzada.

$$\begin{gathered} Y[m,n] = \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} H[k_1,k_2]X[m+k_1-1, n+k_2-1] \\ 1 \leq m \leq M_x - L_1 + 1,\ 1 \leq n \leq N_x - L_2 + 1 \end{gathered} \tag{B.4}$$

Una práctica muy habitual también cuando se trabaja con CNN es emplear un factor de diezmado tanto en el eje horizontal como en el vertical, llamándolos D_1 y D_2, respectivamente. Con esto, la expresión B.3 se transforma en la siguiente:

$$\begin{gathered} Y[m,n] = \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} H[k_1,k_2]X[D_1(m-1)+k_1, D_2(n-1)+k_2] \\ 1 \leq m \leq \left\lfloor \frac{M_x - L_1}{D_1} + 1 \right\rfloor,\ 1 \leq n \leq \left\lfloor \frac{N_x - L_2}{D_2} + 1 \right\rfloor \end{gathered} \tag{B.5}$$

Finalmente, hay un concepto que también conviene comprender cuando se va a trabajar con CNN y este es padding, el cual consiste en añadir ceros alrededor de la imagen para evitar lo que se conoce como efecto borde y que se puede ver como que las componentes exteriores de una imagen contribuyen menos al resultado de la convolución. El padding consiste entonces en añadir P_1 filas encima y P_1 debajo de la imagen, y P_2 columnas a la izquierda y P_2 a la derecha. Así pues, para una señal de entrada con tamaño $M_x \times N_x$, tamaño de filtro $L_1 \times L_2$, factores de diezmado D_1 y D_2 y padding de P_1 y P_2 el tamaño de la señal salida $[M_y, N_y]$ es:

$$[M_y, N_y] = \left[\left\lfloor \frac{M_x + 2P_1 - L_1}{D_1} + 1 \right\rfloor, \left\lfloor \frac{N_x + 2P_2 - L_2}{D_2} + 1 \right\rfloor \right] \tag{B.6}$$

Con todo esto explicado, no es complicado comprender el funcionamiento básico de las redes neuronales convolucionales. Según lo explicado en la sección 2.2 los pesos y sesgos de la red son los parámetros afectan a la entrada y que se actualizan durante el

entrenamiento. Pues bien, en las redes convolucionales (de dos dimensiones) las señales de entrada X son bidimensionales y en lugar de que cada uno de los elementos de la entrada se vea afectado en un principio por una función lineal, es filtrada por un filtro bidimensional H, esto es, se realiza una convolución de X y H. Además, en este caso, los parámetros a optimizar en una red convolucional, en lugar de ser los pesos y sesgos, son los coeficientes de este filtro H. El funcionamiento de las CNN se basa en este principio, siendo además relevante que no se emplea un único filtro en cada una de las capas, sino que se emplea un número α_l de filtros para la capa l, siendo estos números uno de los parámetros del diseño de estos sistemas. Habitualmente, cuando se trata de explicar el motivo por el que se utilizan multitud de filtros se dice que cada uno de ellos se encarga de extraer unas características distintas de la imagen a analizar resultando así más eficiente la optimización global de los coeficientes de todos los filtros empleados.

Con esto, ya es posible definir la salida $Y_l \in \mathbb{R}^{\alpha_l \times M_y \times N_y}$ de la capa l dada una entrada $X_l \in \mathbb{R}^{M_x \times N_x}$ que tiene α_l filtros, por lo que $H_l \in \mathbb{R}^{\alpha_l \times L_1 \times L_2}$ y que emplea una función de activación F quedando finalmente la expresión que da la salida de del filtro k de la capa l $(H_l)_k$:

$$(Y_l)_k[m,n] = F\left(\sum_{k_1=1}^{L_1}\sum_{k_2=1}^{L_2}(H_l)_k[k_1,k_2]X_l[D_1(m-1)+k_1, D_2(n-1)+k_2]\right) \tag{B.7}$$

Otro elemento habitual cuando se trabaja con este tipo de redes es la agrupación o pooling y, aunque puede confundirse con el diezmado, no es lo mismo a pesar de tener un objetivo común, que es la reducción de datos con los que trabajar. Tanto es así que, aunque introducimos este concepto en esta sección por ir unido habitualmente a la utilización de CNN, la realidad es que resulta ser otra capa más no lineal de la red que consiste en convertir una matriz en otra de menor tamaño. Una forma popular de realizar esto es el max-pooling y consiste en agrupar la matriz en submatrices de $p_1 \times p_2$ y escoger el máximo elemento de cada una de estas submatrices tal y como se muestra en la expresión B.8 para casos en los que $p_1 = p_2 = p$, que es lo habitual.

$$(Z_l)_k[m,n] = \max_{\{i,j=1,\dots,p\}} (Y_l)_k[(m-1)p+i, (n-1)p+j] \tag{B.8}$$

El concepto de upsampling realiza la tarea inversa al de pooling y toma también como parámetros de entrada p_1 y p_2.

Como hemos explicado ya, la fase de entrenamiento de estas redes consiste en actualizar los coeficientes de los filtros de forma conceptualmente igual a lo explicado en la sección anterior y cuyo desarrollo no se incluye en este texto por resultar excesivo y no ayudar a la comprensión del mismo.

C. ALGUNOS CONCEPTOS DE TEORÍA DE LA DECISIÓN BAYESIANA

El fin último de este proyecto es realizar un sistema que decida si hay o no una anomalía en un lugar y en un instante concreto. Por eso, en esta sección se dan unas pinceladas básicas de algunas ideas que conviene conocer sobre teoría de la decisión. Este no es un trabajo cuya temática es esta, sino que esto es sólo una herramienta empleada para dar unos resultados mensurables al final de este texto.

El campo de estudio de la teoría de la decisión bayesiana, como cabe esperar, toma como punto de partida el Teorema de Bayes, el cual es uno de los puntos clave en el estudio de las bases de la probabilidad y que relaciona la probabilidad de ocurrencia de un suceso aleatorio H condicionado a la ocurrencia de otro suceso B con la probabilidad de ocurrencia del suceso B condicionado a la ocurrencia del suceso H. Esto, por ejemplo, puede suponer que a partir de la probabilidad de tener dolor de cabeza cuando se tiene gripe se puede calcular la probabilidad de tener gripe cuando se tiene dolor de cabeza. Este teorema se enuncia a continuación.

Sea $H_1, H_2, ..., H_n$ un conjunto de sucesos mutuamente excluyentes y exhaustivos, y tales que la probabilidad de cada uno de ellos es distinta de cero y sea B un suceso cualquiera del que se conocen las probabilidades condicionales $P(B|H_i)$. Entonces, la probabilidad $P(H_i|B)$ viene dada por la expresión:

$$P(H_i|B) = \frac{P(B|H_i)P(H_i)}{P(B)} \tag{C.1}$$

donde:

- $P(H_i)$ son las probabilidades a priori,
- $P(B|H_i)$ es la probabilidad de B en la hipótesis H_i,
- $P(H_i|B)$ son las probabilidades a posteriori.

Para el caso de este trabajo tenemos que hay únicamente dos sucesos ya que el fin no es categorizar las anomalías sino que es exclusivamente detectarlas. Entonces, hacemos que el suceso H_1 se corresponda con la ausencia de eventos anómalos y el suceso H_2 con la presencia de éstos. Para este caso, el análisis de lo que a continuación se explica se simplifica. Además esta explicación se va a realizar acudiendo al caso que nos atañe por fomentar su comprensión e ir preparando al lector para el siguiente capítulo.

Bien, antes de realizar este sistema de detección de anomalías se podrían haber estimado unas probabilidades a priori de normalidad y de anomalía, esto es, $P(H_1)$ y $P(H_2)$, respectivamente donde, por la propia definición de anomalía, es claro que $P(H_1) > P(H_2)$.

Bajo este supuesto y sin la utilización de un sistema de receptor acústico que nos aportara una información adicional, lo que la teoría de la decisión dice es que el suceso ocurrido es siempre H_1. Esto significa pensar que nunca hay anomalías, una hipótesis que desde luego es falsa. Pero esto se soluciona empleando el teorema de Bayes y contando con una información adicional, que es la que nos proporciona el sistema desarrollado en este trabajo. Por esta razón, es necesario buscar un suceso B que nos permita estimar las probabilidades a posteriori $P(H_1|B)$ y $P(H_2|B)$, las cuales serán realmente las utilizadas para tomar la decisión final. Este suceso B está relacionado con la información del receptor acústico procesado a través del sistema de aprendizaje profundo. Sin entrar en detalles aquí, ya que lo haremos en la subsección 3.4.3, pensaremos de momento que el suceso B significa simplemente que el error de predicción es igual a un valor concreto.

Para continuar, surge un nuevo concepto que llamaremos función de coste. Lo que ésta nos indica es cómo de costosa es una acción en relación al suceso ocurrido. Esto es, sean los sucesos $H_1, H_2, ..., H_n$ y las acciones $\alpha_1, \alpha_2, ..., \alpha_r$, entonces $\lambda(\alpha_i|H_j) = \lambda_{ij}$ indica el coste de realizar la acción α_i bajo la ocurrencia del suceso H_j. A partir de esto se define el coste esperado $R(\alpha_i|B)$ de la acción α_i bajo la ocurrencia del suceso B según la expresión:

$$R(\alpha_i|B) = \sum_{j=1}^{n} \lambda_{ij} P(H_j|B) \tag{C.2}$$

En base a esta definición se elige la acción que minimiza este coste esperado, esto es, se elige la acción $\alpha*$ según la expresión:

$$\alpha* = \arg\min_{\alpha_i} R(\alpha_i|B) \tag{C.3}$$

Esto último, aplicado a situaciones con dos sucesos, como es el caso, si además se consideran también dos acciones α_1 y α_2, que se corresponden con decidir el suceso 1 o el 2, respectivamente, se transforma en una expresión cerrada y un umbral que permite decidir si ha ocurrido un suceso u otro. A continuación se explica esto:

$$\begin{aligned} R(\alpha_1|B) &= \lambda_{11} P(H_1|B) + \lambda_{12} P(H_2|B) \\ R(\alpha_2|B) &= \lambda_{21} P(H_1|B) + \lambda_{22} P(H_2|B) \end{aligned} \tag{C.4}$$

Por lo tanto, y en base a lo indicado en C.3, a continuación se desarrolla la expresión que determina bajo qué condición se ha de elegir la acción α_1, es decir, interpretar que ha ocurrido el suceso H_1:

$$\begin{gathered} R(\alpha_1|B) < R(\alpha_2|B) \leftrightarrow (\lambda_{21} - \lambda_{11}) P(H_1|B) > (\lambda_{12} - \lambda_{22}) P(H_2|B) \leftrightarrow \\ (\lambda_{12} - \lambda_{22}) P(B|H_1) P(H1) > (\lambda_{21} - \lambda_{22}) P(B|H_2) P(H2) \leftrightarrow \\ \frac{P(B|H_1)}{P(B|H_2)} > \frac{\lambda_{12} - \lambda_{22}}{\lambda_{21} - \lambda_{11}} \frac{P(H_2)}{P(H_1)} \end{gathered} \tag{C.5}$$

La fuerza de esta expresión es que establece un umbral que es independiente de la observación B. Además, es también habitual imponer que $\lambda_{11} = \lambda_{22} = 0$ debido a que se

considera que un acierto no tiene coste, por lo que se llega a una expresión reducida que indica bajo qué condición decidir H_1 y H_2:

$$\begin{aligned} \text{Elegir } H_1 \text{ si } \frac{P(B|H_1)}{P(B|H_2)} &> \frac{\lambda_{12}}{\lambda_{21}} \frac{P(H_2)}{P(H_1)} \\ \text{Elegir } H_2 \text{ si } \frac{P(B|H_1)}{P(B|H_2)} &\leq \frac{\lambda_{12}}{\lambda_{21}} \frac{P(H_2)}{P(H_1)} \end{aligned} \tag{C.6}$$

La expresión anterior, por tanto, nos da una forma cerrada de decidir si el suceso ocurrido es H_1 o H_2, ya que el término de la izquierda de la inecuación es lo que se conoce como el ratio de verosimilitud y suele calcularse modelando $P(B|H_1)$ y $P(B|H_2)$ mediante funciones de probabilidad conocidas y, por otro lado, el término de la derecha se conoce como umbral de decisión y, como ya hemos comentado, no depende de la observación B, por lo que es fijo para un escenario concreto.

Ya para terminar, a la hora de calificar el rendimiento de un sistema se emplean los términos que se explican en la tabla C.1 y que están íntimamente relacionados con las funciones de coste λ_{ij}.

Concepto	Suceso ocurrido	Suceso detectado	Función de coste relacionada
Verdadero negativo	H_1	H_1	λ_{11}
Falso negativo	H_1	H_2	λ_{12}
Falso positivo	H_2	H_1	λ_{21}
Verdadero positivo	H_2	H_2	λ_{22}

TABLA C.1. CONCEPTOS PARA MEDIR RENDIMIENTO DE UN DETECTOR

Y a partir de estos conceptos se describen otros dos que sirven para entender de forma rápida cómo es el rendimiento de un sistema de detección. Estos dos conceptos son la probabilidad de detección (P_d) y probabilidad de falsa alarma (P_{fa}). Aunque se definen matemáticamente en la expresión C.7, para comprender de forma intuitiva lo que significan en nuestro caso, P_d es la probabilidad de detectar una anomalía cuando la hay y P_{fa} la probabilidad de que en caso de no haber una anomalía el detector considere que sí la hay. Cuando se diseña un sistema de detección se ha de llegar a un compromiso entre estos dos valores, pues siempre se puede mejorar uno a costa de empeorar el otro. Si se quiere una probabilidad de detección alta también se correrá el riesgo de que haya más falsas alarmas, y viceversa.

$$\begin{aligned} P_d &= \frac{VP}{VP + FN} \\ P_{fa} &= \frac{FP}{FP + VN} \end{aligned} \tag{C.7}$$

En base a estos dos términos, surge otro más, que es el de curva ROC (Receiver operating characteristic) y que es simplemente una gráfica que tiene en su eje de abcisas el valor

de P_{fa} y en su eje de ordenadas en valor de P_d. Estas curvas son especialmente interesantes porque permiten analizar de manera rápida el compromiso entre las probabilidades de detección y falsa alarma.

En la figura C.1 se muestran unas curvas ROC que corresponden a tres sistemas A, B y C con distintas prestaciones. El A es el que peor rendimiento proporciona resultando no informativo y es el peor caso posible de curva ROC. El sistema B es uno realista y en el que se ve bien el compromiso entre probabilidades de falsa alarma y detección. Por último, el sistema C da el rendimiento perfecto, ya que se puede conseguir una probabilidad de detección de 1 y una de falsa alarma de 0 de forma simultánea.

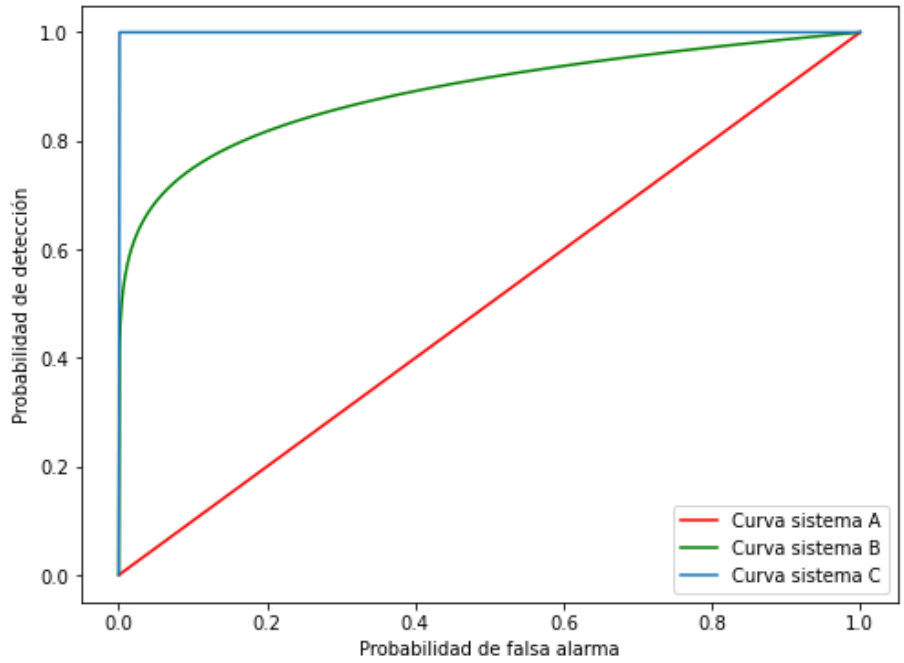


Fig. C.1. Curvas ROC que describen el rendimiento de tres sistemas con distintas prestaciones

D. FICHAS DE LAS MEDIDAS

Etiqueta medida	Comentario	Duración total	Tipo de evento	Duración (s)
Entra	Excavadora avanza	300	Avanza	106
			Silencio	194
Oruga	Distancia = 0 m	240	Silencio	60
			Anda	39
			Silencio	40
			Anda	50
			Silencio	51
Oruga	Distancia = 10 m	240	Silencio	60
			Anda	43
			Silencio	46
			Anda	45
			Silencio	46
Cazo	Distancia = 0 m	240	Silencio	50
			Cava	40
			Silencio	48
			Tapa	66
			Silencio	36
Martillo Hidráulico	Iteración 1	120	Silencio	42
			Pico 0 m	11
			Silencio	29
			Pico 5 m	11
			Silencio	18
			Pico 10 m	9

TABLA D.1. FICHAS DE LAS MEDIDAS CON EVENTOS UTILIZADAS PARA DESARROLLAR Y MEDIR EL RENDIMIENTO DE ESTE TRABAJO